STANFORD UNIVERSITY

CENTER FOR SYSTEMS RESEARCH

Optimal Regulation of Physiological Systems via Real-Time Adaptive Model Synthesis

by

Cristy M. Schade

Technical Report No. 6792-2

Information Systems Laboratory

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
Stanford Electronics Laboratories
Stanford University, Stanford, California

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP
NA

3. REPORT TITLE
OPTIMAL REGULATION OF PHYSIOLOGICAL SYSTEMS VIA REAL-TIME ADAPTIVE MODEL SYNTHESIS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Technical Report

5. AUTHOR(S) (First name, middle initial, last name)
Cristy M. Schade

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
August 1971	87	49

8a. CONTRACT OR GRANT NO.
Joint Services Electronics Program
b. PROJECT NO. N00014-67-A-0112-0044
c.
d.

9a. ORIGINATOR'S REPORT NUMBER(S)
TECHNICAL REPORT NO. 6792-2

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
SU-SEL-71-033

10. DISTRIBUTION STATEMENT Reproduction in whole or in part permitted for any purpose of US Govt. Has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
U.S. Army, U.S. Navy and U.S. Air Force

13. ABSTRACT
Optimal control algorithms that use an adaptive non-recursive digital filter model for on-line closed-loop blood pressure regulation have been developed. This Automatice Therapeutic Control System was designed specifically for the regulation of physiological systems, but the design assumptions are such that it should prove very useful for a much broader class of control problems. An Adaptive Model Control system has been developed, the analysis of which is presented in two parts: (1) the real-time adaptive model synthesis procedure, and (2) the optimal forward-time controller. The Adaptive modeling process is accomplished by the rapidly converging α-LMS Algorithm. The conditions necessary to guarantee convergence for deterministic inputs are presented for zero mean and nonzero mean additive-output noise systems and also for the low-order approximation problem. The optimal forward-time controller is described. It not only makes efficient use of the mathematical properties of the non-recursive digital filter model (that is, the filter gains and the state of the filter), but also meets the time and memory constraints of a mini-computer used on-line for this control problem. The future control inputs are determined by a doubly constrained quadratic function which is solved to minimize the mean-square control error. The results of an experimental run are included in which these algorithms were used to regulate the blood pressure of a dog that had been artificially placed in a hypotensive state (shock). The results of this and similar experiments have been very successful from both an engineering and medical point of view, and if the necessary arrangements can be made with the local hospitals, this system will be used in the near future as part of an intensive care unit.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
ADAPTIVE CONTROL						
BIOMEDICAL ENGINEERING						
LEARNING SYSTEMS						

SEL-71-033

OPTIMAL REGULATION OF PHYSIOLOGICAL SYSTEMS
VIA REAL-TIME ADAPTIVE MODEL SYNTHESIS

by
Cristy M. Schade

August 1971

Technical Report No. 6792-2

This work was partially supported by the
Joint Services Electronics Program
(U.S. Army, U.S. Navy, and U.S. Air Force)
under Contract N00014-67-A-0112-0044

Information Systems Laboratory
Stanford Electronics Laboratories
Stanford University Stanford, California

ABSTRACT

Optimal control algorithms that use an adaptive non-recursive digital filter model for on-line closed-loop blood pressure regulation have been developed. This Automatic Therapeutic Control System was designed specifically for the regulation of physiological systems, but the design assumptions are such that it should prove very useful for a much broader class of control problems.

An Adaptive Model Control system has been developed, the analysis of which is presented in two parts: (1) the real-time adaptive model synthesis procedure, and (2) the optimal forward-time controller.

The Adaptive modeling process is accomplished by the rapidly converging α-LMS Algorithm. The conditions necessary to guarantee convergence for deterministic inputs are presented. Mean-Square Error bounds are presented for zero mean and nonzero mean additive-output noise systems and also for the low-order approximation problem.

The optimal forward-time controller is described. It not only makes efficient use of the mathematical properties of the non-recursive digital filter model (that is, the filter gains and the state of the filter), but also meets the time and memory constraints of a minicomputer used on-line for this control problem. The future control inputs are determined by a doubly constrained quadratic function which is solved to minimize the mean-square control error.

The results of an experimental run are included in which these algorithms were used to regulate the blood pressure of a dog that had been artificially placed in a hypotensive state (shock). The results

of this and similar experiments have been very successful from both an engineering and medical point of view, and if the necessary arrangements can be made with the local hospitals, this system will be used in the near future as part of an intensive care unit.

ACKNOWLEDGMENT

Professor Otto Smith of the University of California at Berkeley introduced me to adaptive control systems, and Noel Thompson, M.D., of the Palo Alto Medical Research Foundation introduced me to the theory of Automatic Therapeutic Control Systems. During the course of this research I had helpful discussions with D. Austin, J. Bliss, J. Delaney, G. Franklin, O. Frost III, B. Goode, T. Kailath, G. Sidhu, and B. Widrow. I am grateful to J. Delaney, L. Kaufman, F. Mansfield, T. Stibolt Jr., and many others who helped with the computer programming and on-line experiments. Mrs. N. Wynkoop typed the manuscript.

Special thanks go to my wife Kathleen who has been a continual source of happiness and inspiration throughout the course of this research.

The financial support provided by my National Science Foundation Fellowship, Stanford Fellowship and Tau Beta Pi Fellowship and contract N00014-67-A-0112-0044 is gratefully acknowledged.

CONTENTS

TABLES

ILLUSTRATIONS

1. INTRODUCTION

A. AUTOMATIC THERAPEUTIC CONTROL SYSTEMS

Therapeutic is defined as that which is related to the branch of medicine dealing with the treatment of disease [1]. Thus, an automatic therapeutic control system is any control syst~~e~~m which deals with the treatment of an illness, sickness, interruption or perversion of function of any of the organs in an automatic (closed loop) manner. The automatic therapeutic control system proposed in this paper, an adaptive model controller (see Fig.7), was used for the treatment of shock (hypotension). For this problem a pressure-elevating drug was used to regulate the average blood pressure of a dog which had been placed in a controlled state of shock.

Another promising area of application is anesthesiology. Many of the chemicals used in anesthesia, both in the liquid and in the gaseous form, are sufficiently fast acting to utilize fully this control system. This system for quickly controlling the degree to which a patient is anesthetized would thus be a great help to an anesthesiologist.

B. THE CARDIOGENIC SHOCK PROBLEM

It is estimated that in the United States alone 250,000 lives are lost annually because of cardiogenic shock [2,3]. The best treatment for this deadly problem is still disputed, and it is claimed that cardiogenic shock still defies the cardiologist's skill 80 to 90 per cent of the time.

Thus, there is a need for new and improved ways to deal with cardiogenic shock. One possible method of treatment is to regulate automatically a patient's blood pressure by administering drugs with a control system. The engineering control problems associated with this approach are what we are concerned with in this paper.

The methods of treatment for cardiogenic shock can presently be broken into two opposing schools of thought: those who prefer drug therapy and those who prefer mechanical intervention, particularly with the intra-aortic balloon pump. We are concerned with the drug therapy approach, which can be broken into three schools of thought [2].

> "The vasopressor school, which is becoming popular again, emphasizes the need for agents which increase coronary perfusion pressure. But proponents of vasodilators contend that the major need is to relieve intense vasoconstriction. This mode of therapy reduces the work load of the heart and redistributes blood flow to all areas of the body. Still a third group emphasizes the need for increased myocardial contractility to restore cardiac output."

Regardless of the approach, the objective is to take therapeutic measures early enough to prevent irreversible shock from developing. Irreversible shock can be described as a state of positive feedback where the failure of one function causes still further failure of another function within the same control loop [4,5]. Figure 1 is a simplified block diagram of some of the different types of feedback that can lead to a progressive state of shock (irreversible shock)[5].

We again remind the reader that this paper is concerned with description and analysis of a control system which regulates blood pressure; we do not address the larger problem of what therapy is best for cardiogenic shock.

C. THE COMPUTER LABORATORY & OPERATING ROOM FACILITIES

The computer laboratory used in this research is located on the Stanford Campus in the Durand Laboratory, and the operating room is

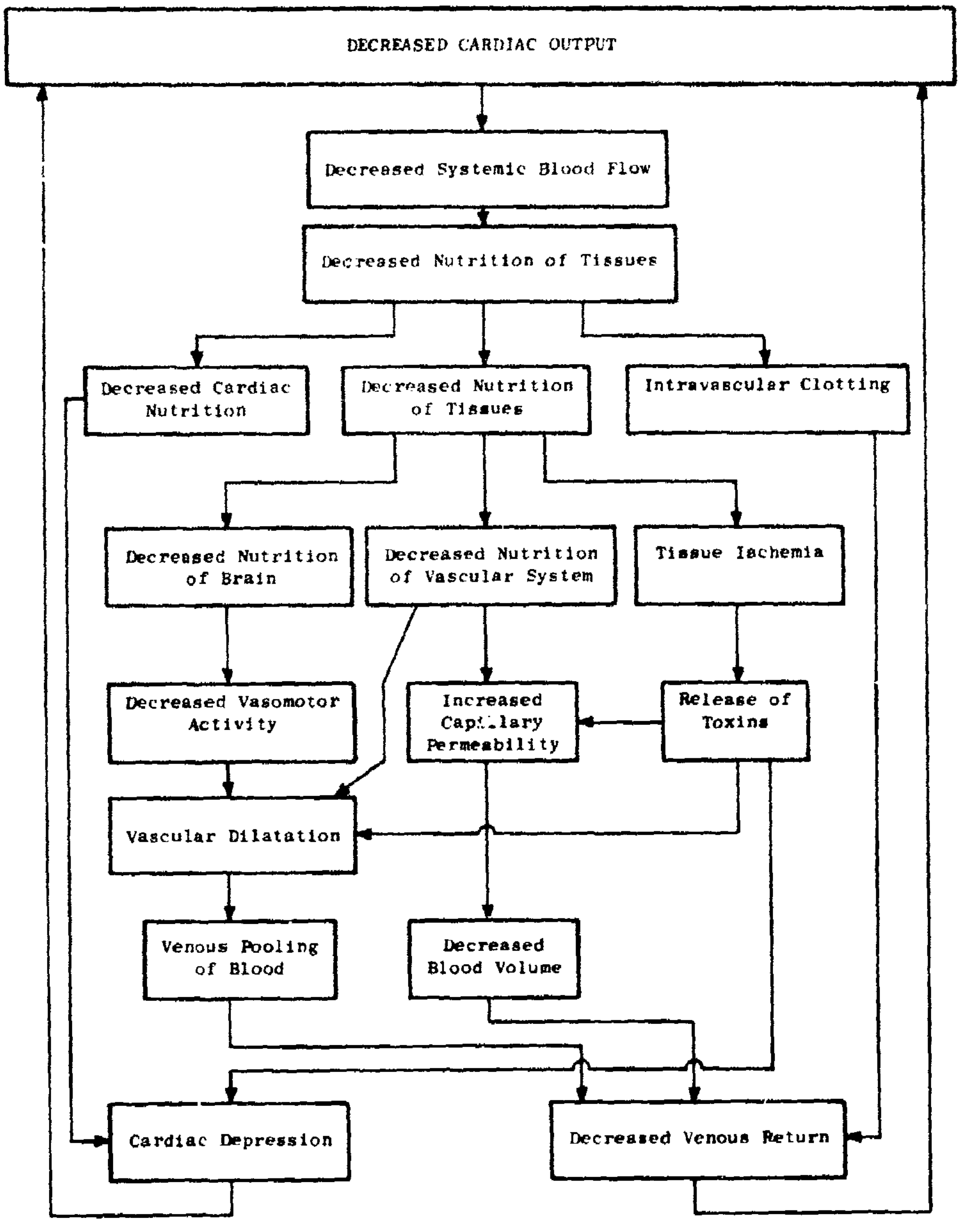

DECREASED CARDIAC OUTPUT
Decreased Systemic Blood Flow
Decreased Nutrition of Tissues
Decreased Cardiac Nutrition
Decreased Nutrition of Tissues
Intravascular Clotting
Decreased Nutrition of Brain
Decreased Nutrition of Vascular System
Tissue Ischemia
Decreased Vasomotor Activity
Increased Capillary Permeability
Release of Toxins
Vascular Dilatation
Venous Pooling of Blood
Decreased Blood Volume
Cardiac Depression
Decreased Venous Return

Fig. 1. Different Types of Feedback That Can Lead to Progression of Shock.

located in Palo Alto at the Palo Alto Medical Research Foundation. A voice grade telephone line is used as the data link between the two facilities (see Fig. 2). This choice of data link will make it very inexpensive and easy to reach patients in the intensive care units at any of the local hospitals.

The primary hardware components are shown in block diagram form in Fig. 3. In the operating room, the strain gauge and drop master* are commercially available items, while the voltage-controlled oscillator (VCO) and drop-rate DETECTOR were designed and built at Stanford. The acoustic couplers and the HP-2116B computer system are commercially available items, but the frequency-to-digital converter (FDC) and the drop-rate ENCODER were also designed and built at Stanford.

The frequency band from 1KHz to 3 KHz is used for transmission of the instantaneous blood pressure. The remaining channel space of the telephone line (500 to 1000 Hz) is used to transmit the encoded drop rate commands. A tone-burst code is transmitted each time a drop of drug is required.

The HP-2116B computer is a mini-computer with a 16,000 word (16 bits per word) memory and a 1.6 microsecond cycle time. In addition to the major components shown in Fig. 3, this computer system has a real time clock which is required for scheduling data gathering, data processing (the forward time calculations and adaptive algorithms) and data output.

*A drop master is a machine which releases one drop of fluid in response to either an internal timer or an external command.

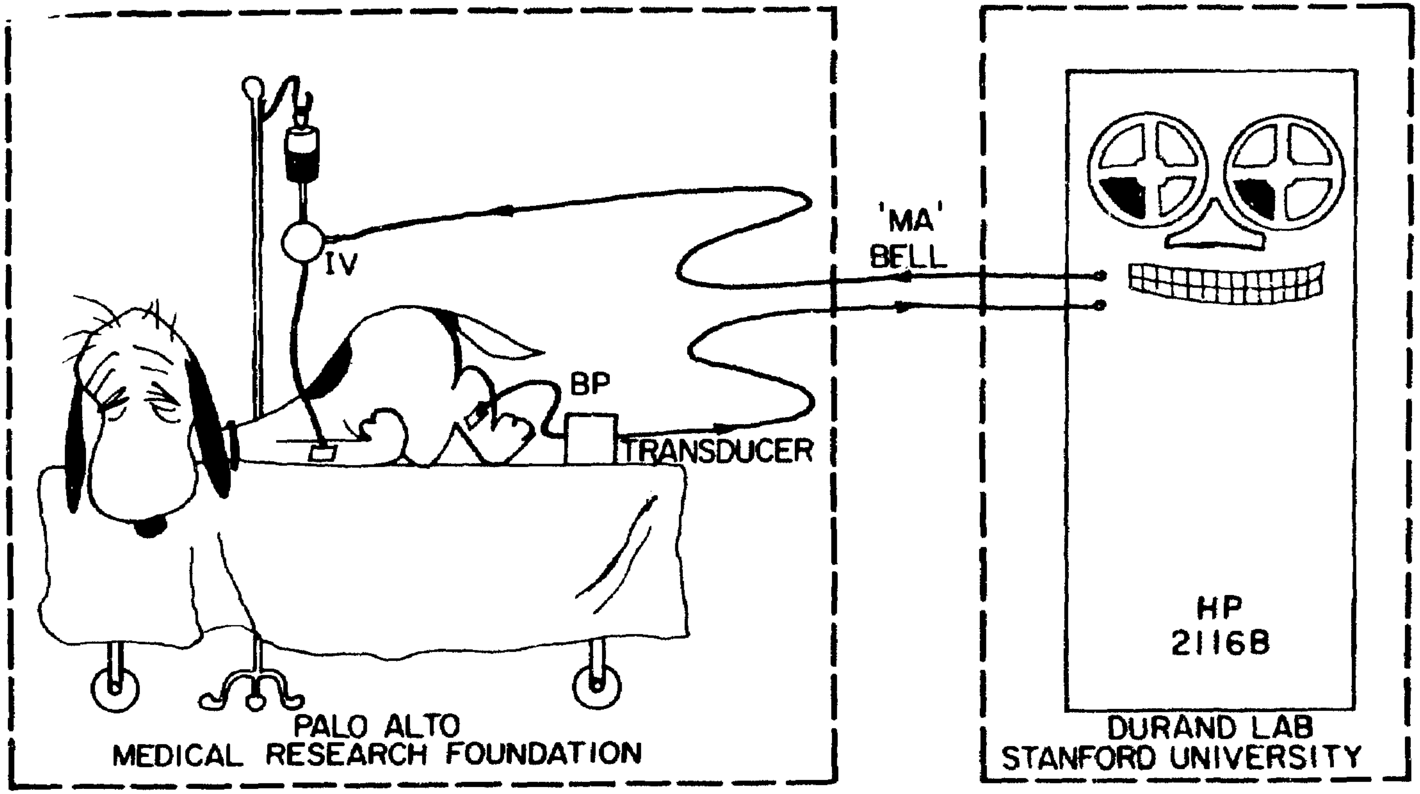


Fig. 2. Pictorial Representation of the Experimental System.

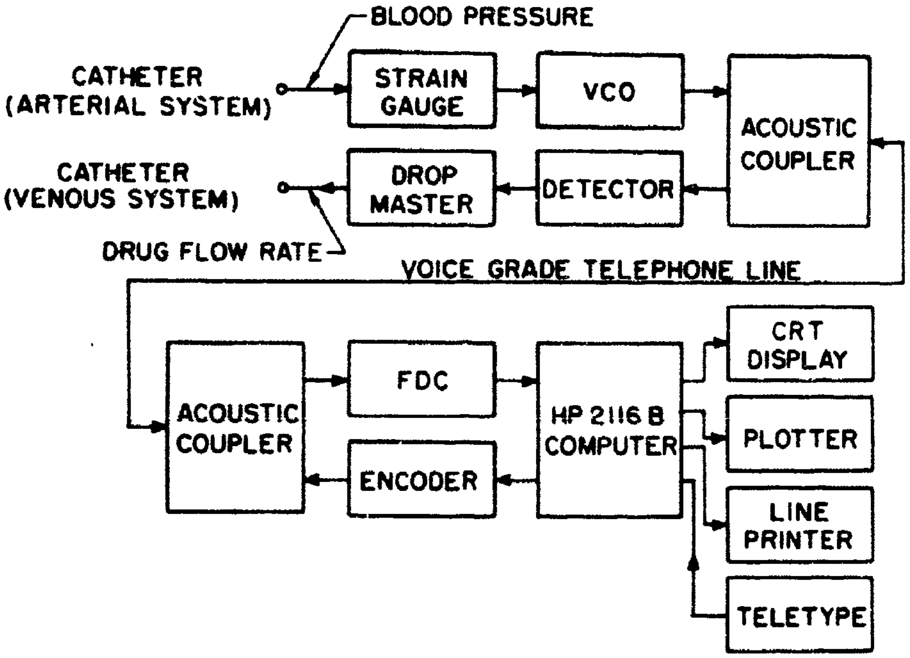


Fig. 3. Block Diagram of the Primary Hardware Components.

Interested readers are referred to references [6-8] for a more complete description of the biomedical engineering problems and requirements associated with patient monitoring and automatic therapeutic control systems.

D. THE ADAPTIVE MODEL CONTROLLER

When designing controllers for real physical systems, we can cope with only those problems that can be foreseen and allowed for in advance. Thus, the idea of an adaptive control system which can compensate for the time-varying and nonlinear aspects of a system is very attractive. The adaptive control principle in essence consists of three things [9]:

1. The definition of an optimum condition of operation.
2. The comparison of the actual performance with the desired performance.

3. The adjustment of system parameters by means of closed-loop operation so as to drive the actual performance toward the desired performance.

One of the simplest systems which employs these principles is the adjustable compensator controller (see Fig. 4). The adaptive filter is adjusted to give, say, a minimum mean-square error. The problem with this system is that the desired response d_j which is needed to adapt the filter is unknown. Note that if d_j were known, then the adaptive filter would be unnecessary.

A more workable adaptive controller is the inverse-model controller (see Fig. 5). Under the appropriate conditions this system is very useful [10]. Its major shortcomings are due to the fact that we must have an estimate for the delay Z^{-N} and the inverse model $\underline{G}^{-1}$ must be well defined.

Another approach is the feedback model controller (see Fig. 6) which uses a faster-than-real-time closed-loop model of the physical system to adapt the compensator. While this is certainly attractive because we can exercise the upper loop in order to adapt the compensator without disturbing the physical system, the problem lies in the fact that the mean-square-error performance function is knownto be irregular, non-parabolic, and containing relative optima [11].

The problems inherent in the above-mentioned systems are overcome by the proposed Adaptive Model Controller of Fig. 7. In this system the identification process is achieved by the rapidly converging α-LMS Algorithm, and the adjustable controller is a faster-than-real-time forward-time calculation which minimizes the squared error $(r_j - y_j)^2$. In this system the control is optimal in a squared-error sense if the

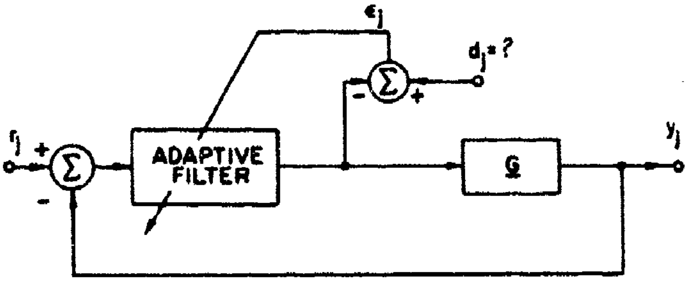


Fig. 4. Adjustable Compensator Controller.

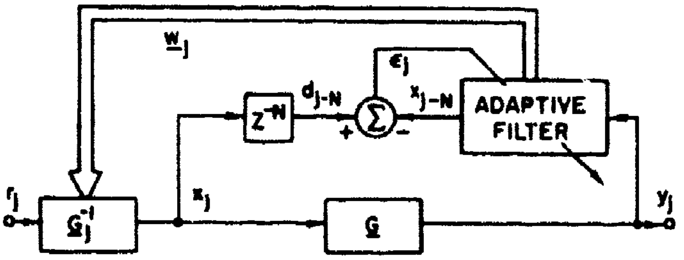


Fig. 5. Inverse-Model Controller.

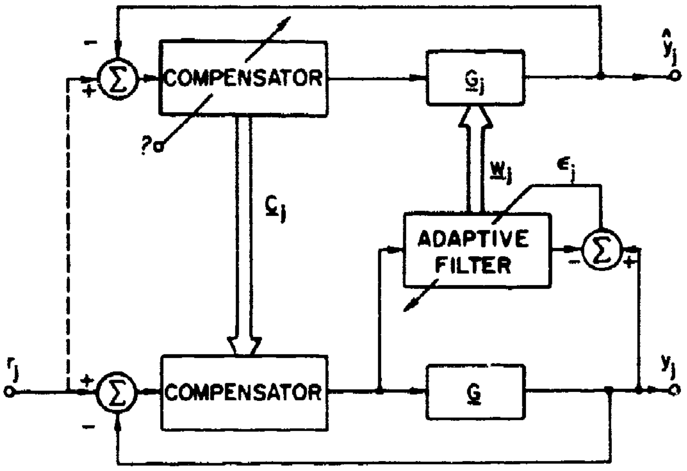


Fig. 6. Feedback Model Controller.

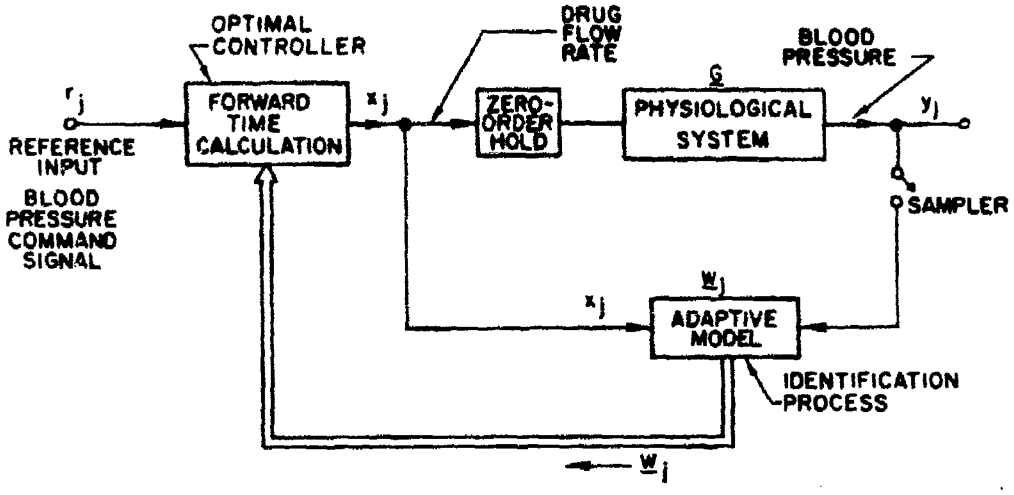


Fig. 7. Adaptive Model Controller.

adaptive model has converged.* Thus, in practice the system is run open loop until the adaptive model has converged, and then the loop is closed through the forward-time calculation.

We will use the following notation to distinguish between scalar, vector and matrix variables:

1. a variable with no underlining is a scalar,
2. a variable with a single underline is a vector, and
3. a variable with a double underline is a matrix.

For example (see Fig. 8) x_j is scalar input at time j, and $\underline{x}_j$ is the vector "input" at time j.

In addition, we will adopt the convention that the weight vector $\underline{w}_j$ (see Fig. 8) is of length N.

*The convergence proofs given in section 2 assume that the physiological system is time-invariant.

2. REAL-TIME ADAPTIVE MODEL SYNTHESIS

A. WIDROW-HOFF ALGORITHM (α-LMS ALGORITHM)

The theory of least-mean-square error filter design has received a considerable amount of attention in the last two decades. The theory has been extended from the problems of filtering and prediction [12] to include those of system identification, process control, and pattern recognition. The Widrow-Hoff Algorithm [13] (see Fig. 8.)

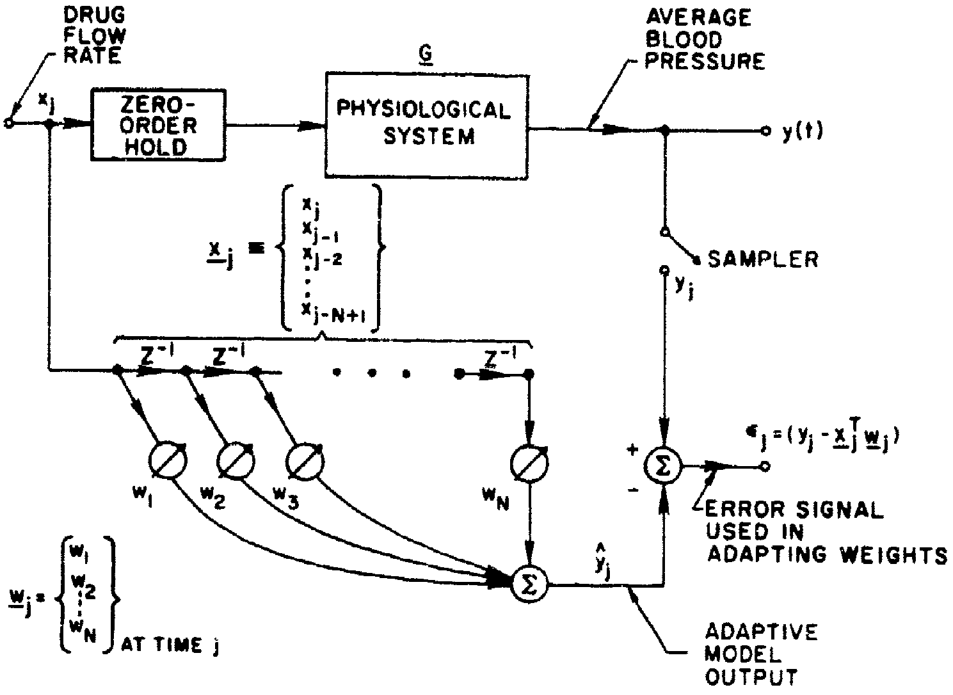
$$\underline{w}_{j+1} = \underline{w}_j + \frac{\alpha}{\|\underline{x}_j\|^2} \underline{x}_j \epsilon_j \tag{2.1}$$

where

$$\epsilon_j = y_j - \underline{x}_j^T \underline{w}_j$$

was originally proposed for use in systems where little or no a priori statistical information is available and where memory size and computational speed are limited. The theory of the α-LMS Algorithm and its many variants has been pursued diligently during the last decade [14-28]. But the behavior of the α-LMS Algorithm in the presence of highly correlated or dependent inputs has not been analyzed. The previous methods of proof basically required that $E[\underline{x}_j \underline{x}_j^T]$ be nonsingular, and that the sequential input vectors be uncorrelated. The need to alleviate these assumptions is obvious in any system where there is a real-time identification utilized in a larger control process.

The behavior of the α-LMS Algorithm when subjected to deterministic inputs is the thesis of this section. Motivated by the conditions on deterministic inputs discovered by Spain [29] for identification and modeling of discrete linear systems, we are able to derive a related


DRUG FLOW RATE
x_j
ZERO-ORDER HOLD
G
PHYSIOLOGICAL SYSTEM
AVERAGE BLOOD PRESSURE
y(t)
SAMPLER
y_j
$\underline{x}_j \equiv \{x_j, x_{j-1}, x_{j-2}, \ldots, x_{j-N+1}\}$
Z^{-1}
Z^{-1}
Z^{-1}
w_1
w_2
w_3
w_N
Σ
$\hat{y}_j$
+
-
Σ
$\epsilon_j = (y_j - \underline{x}_j^T \underline{w}_j)$
ERROR SIGNAL USED IN ADAPTING WEIGHTS
ADAPTIVE MODEL OUTPUT
$\underline{w}_j = \{w_1, w_2, \ldots, w_N\}$ AT TIME j

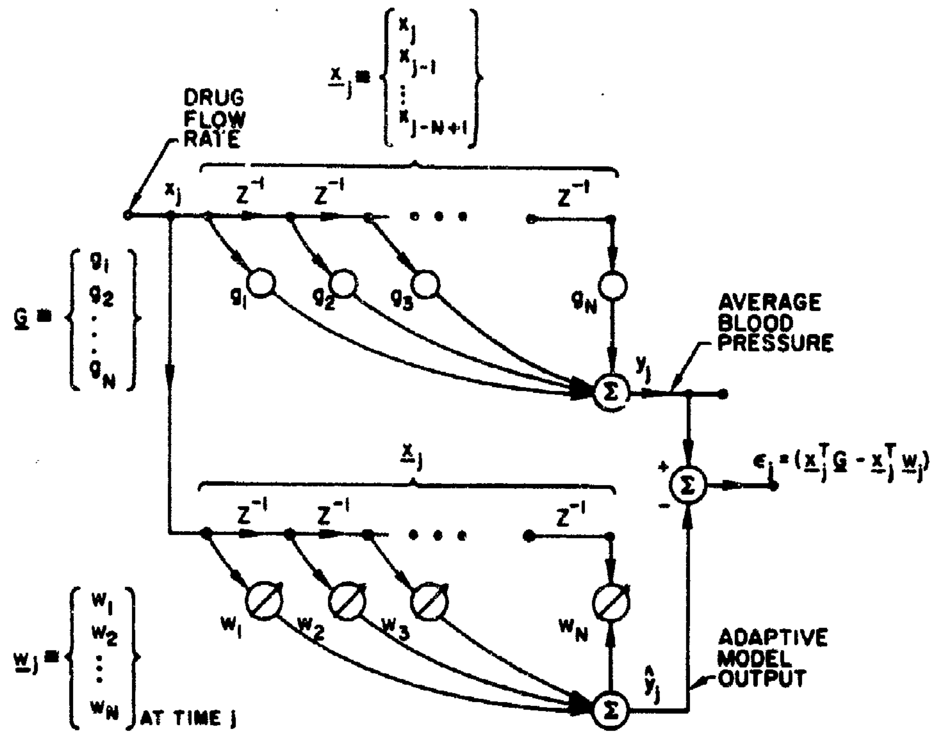
$$\underline{w}_{j+1} = \underline{w}_j + \frac{\alpha}{\|\underline{X}_j\|^2} \underline{X}_j \epsilon_j$$

Fig. 8. Details of the Identification System, an N-tap Non-recursive Adaptive Filter.

condition which we call an "N-dimensional input sequence". The method of proof was motivated by Kailath's innovations approach [30,33] in which the input is transformed into an orthogonal sequence of vectors which represent the 'new' information at each instant of time. The proof is then generalized to include the effects of additive output noise and high and low order approximation.

B. STABILITY OF THE α-LMS ALGORITHM

In order to gain some analytical insight as to how the α-LMS Algorithm behaves when used as an identification process, let us represent the physiological system to be modeled by a non-recursive discrete filter $\underline{G}$ of length N. (See Fig. 9.) The motivation for this representation is twofold: First, the filter gains g_i, i=1,2,...N are equal to the impulse response of the physiological system at times $(i-1)\Delta$ where Δ equals the time delay between each filter tap. Thus, from our experimental data we choose N and Δ such that $(N-1)\Delta$ equals the total time of the impulse response and such that Δ is small enough to give a "good" piecewise-linear approximation to the continuous impulse response. Second, this representation gives us a one-to-one correspondence between the elements of the filter $\underline{G}$ and the elements of the weight vector $\underline{w}_j$. Thus, the modeling error at time j, ϵ_j,



$\underline{x}_j = \{x_j, x_{j-1}, \vdots, x_{j-N+1}\}$
DRUG FLOW RATE
x_j
z^{-1}
z^{-1}
z^{-1}
$\underline{G} = \{g_1, g_2, \vdots, g_N\}$
g_1
g_2
g_3
g_N
AVERAGE BLOOD PRESSURE
y_j
Σ
$\underline{x}_j$
+
-
$\epsilon_j = (\underline{x}_j^T \underline{G} - \underline{x}_j^T \underline{w}_j)$
z^{-1}
z^{-1}
z^{-1}
$\underline{w}_j = \{w_1, w_2, \vdots, w_N\}$ AT TIME j
w_1
w_2
w_3
w_N
ADAPTIVE MODEL OUTPUT
$\hat{y}_j$

Fig. 9. Idealized System Where the Physiological System is Replaced by Its Impulse Response.

can be written as follows

$$\begin{aligned}\epsilon_j &= y_j - \underline{x}_j^T \underline{w}_j \\ &= \underline{x}_j^T \underline{G} - \underline{x}_j^T \underline{w}_j \\ &= \underline{x}_j^T (\underline{G} - \underline{w}_j) \qquad (2.2)\end{aligned}$$

As a worst-case analysis, we would like to know the worst that can happen to $\|\underline{G} - \underline{w}_j\|$ when $\underline{x}_j$ is arbitrary, i.e., is the α-LMS Algorithm stable? First, we consider the degenerate case of $\underline{x}_j \equiv 0$. By definition, if $\underline{x}_j \equiv 0$, then (2.1) is $\underline{w}_{j+1} = \underline{w}_j$. Note also that if $\underline{x}_j \equiv 0$, then by (2.2) $\epsilon_j = 0$. The general result is that without making any assumptions about the input, we have:

Theorem 1. Given a non-recursive filter $\underline{G}$ of length N and any input $\underline{x}_j$ then the next weight vector $\underline{w}_{j+1}$ obtained by α-LMS adaption $(0 < \alpha < 2)$ is such that

$$\|\underline{w}_{j+1} - \underline{G}\| \leq \|\underline{w}_j - \underline{G}\| \qquad (2.3)$$

where equality holds iff $\epsilon_j = 0$.

Theorem 1 is proved in Appendix A.

Theorem 1 states that no matter what the input $\underline{x}_j$ is, the distance between $\underline{w}_j$ and $\underline{G}$ is non-increasing. Thus, referring to Fig. 10, we can see that the tip of the weight vector $\underline{w}_{j+1}$ will always be contained inside some hypersphere of diameter $\|\underline{w}_j - \underline{G}\|$ centered on the tip of $\underline{G}$. In other words, (2.1) is stable (convergent) for $0 < \alpha < 2$.

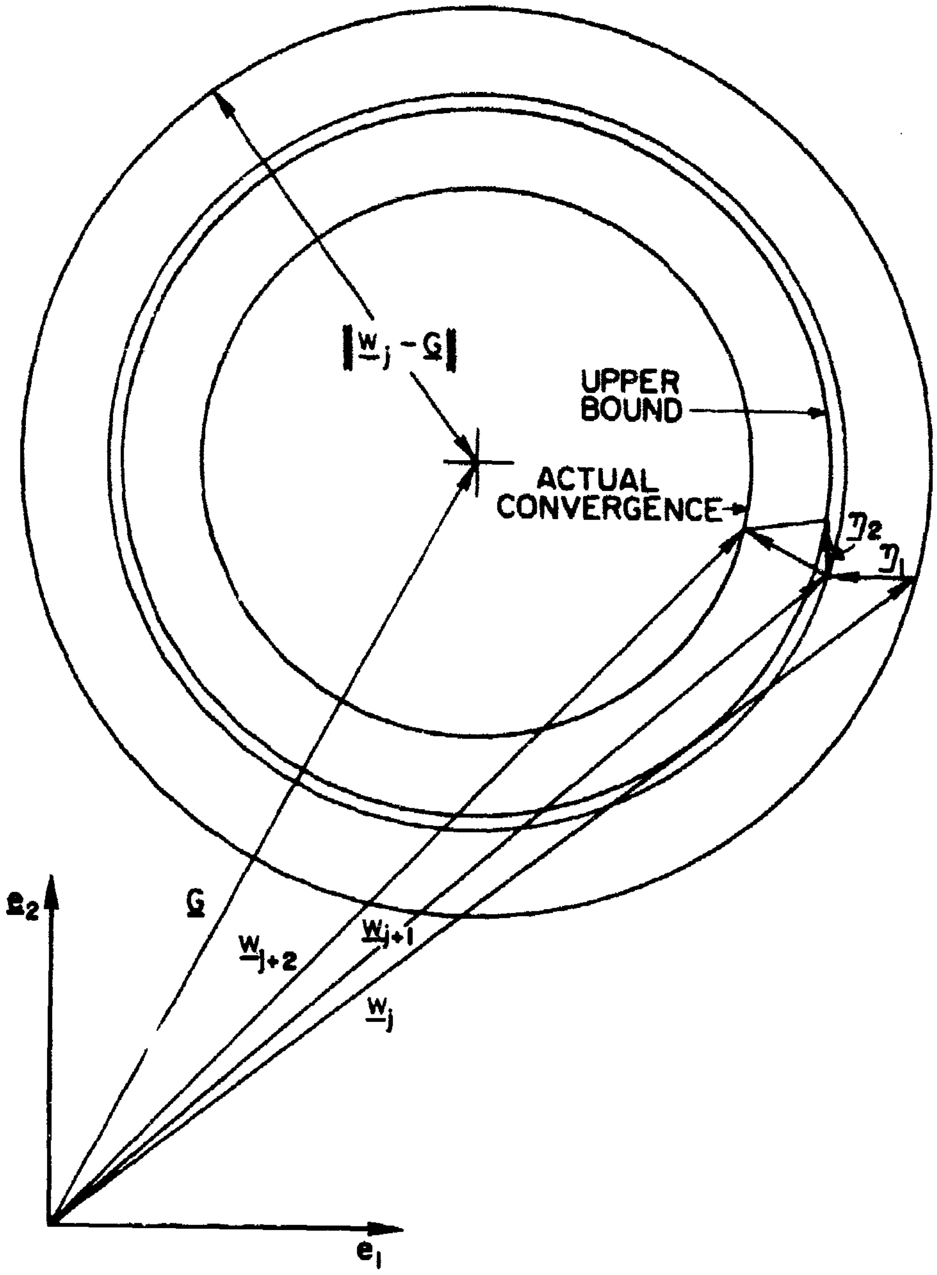


Fig. 10. An Illustration of the Upper Bound for N = 2.

C. UNBIASED CONVERGENCE

Before introducing the definition of an N-dimensional input sequence, let us examine some of the more obvious properties of the algorithm (2.1). Take the case where $\alpha = 1$ and the input sequence is

$$\begin{aligned}
\underline{x}_1^T &= [100 \ldots 0] \\
\underline{x}_2^T &= [0100 \ldots 0] \\
&\vdots \qquad \vdots \\
\underline{x}_N^T &= [0 \ldots 001] \\
\underline{x}_{N+1}^T &= [1000 \ldots 0] \\
&\vdots \qquad \vdots
\end{aligned}$$

then it is easy to show that given any finite $\underline{w}_o$, we have $\underline{w}_j = \underline{G}$ for all $j \geq N + 1$. Furthermore, if $\alpha \neq 1$ but $0 < \alpha < 2$, then it is not hard to show that $\lim_{j \to \infty} \underline{w}_j = \underline{G}$.

There are many such examples that demonstrate the unbiased convergence of (2.1), but what can be said about an arbitrary input sequence $\{\underline{x}_j\}_o^\infty$? So that we may deal with such a question in a rigorous manner, we let $\underline{x}_j$ be the present input vector, an arbitrary vector in an Euclidean space E^N , and let S be a subspace containing the previous $N - 1$ vectors, $\underline{x}_{j-1}, \underline{x}_{j-2}, \ldots, \underline{x}_{j-N+1}$. Then the present vector $\underline{x}_j$ can be uniquely represented in the form [34]

$$\underline{x}_j = \underline{\eta}_j + \tilde{\underline{x}}_j$$

where

$$\underline{\eta}_j \perp S \qquad \text{and} \qquad \tilde{\underline{x}}_j \in S \quad .$$

We can now make the following formal definition of an N-dimensional input sequence:

Definition. An N-dimensional input sequence is any sequence of vectors $\{\underline{x}_j\}_o^\infty$ where a sequence member is

$$\underline{x}_j \triangleq \begin{bmatrix} x_j \\ x_{j-1} \\ \vdots \\ x_{j-N+1} \end{bmatrix}$$

such that (1) the sequence of vectors $\{\underline{x}_i\}_{j-N}^{j-1}$ has N linearly independent element vectors for all $j \geq 2N$, and (2) there exists a $0 < \delta \leq 1$ such that

$$\frac{\|\underline{\eta}_j\|^2}{\|\underline{x}_j\|^2} \geq \delta \qquad \text{for all} \qquad j \geq N \quad .$$

The restrictions imposed by this definition will be relaxed in a subsequent development.

Consider condition (1) with $N = 2$. We have a matrix, $\underline{\underline{B}}_j$, say, where

$$\underline{\underline{B}}_j = \begin{bmatrix} \underline{x}_{j-1}^T \\ - - - \\ \underline{x}_j^T \end{bmatrix} = \begin{bmatrix} x_{j-1} & x_{j-2} \\ x_j & x_{j-1} \end{bmatrix}$$

At time j, the scalar x_j is the only new input value, and if $\det \underline{\underline{B}}_j \neq 0$ then we must have one of two cases: if $x_{j-2} = 0$ then x_j is arbitrary, else

$$x_j \neq \frac{x_{j-1}^2}{x_{j-2}} \quad .$$

Condition (2) excludes a set of values x_j at time j which would produce vectors $\underline{x}_j$ in E^N whose boundary defines a circular hypercone centered at the origin with its axis perpendicular to S. To see this note that

$$\frac{\|\underline{\eta}_j\|^2}{\|\tilde{\underline{x}}_j\|^2 + \|\underline{\eta}_j\|^2} \geq \delta$$

and thus

$$\|\underline{\eta}_j\|^2 \geq \frac{\delta}{1 - \delta} \|\tilde{\underline{x}}_j\|^2 \quad .$$

A constant input vector is f)rbidden if we are to satisfy the definition of an N-dimensional input sequence. It is obvious that we wish to exclude such a sequence, for one cannot measure system dynamics without perturbing the system. The fact that $\underline{\underline{B}}_j$ must have rank N corresponds to the fact that there are N states in the filter and all states must be excited to be identified. Equivalently, the filter spans E^N therefore $\underline{\underline{B}}_j$ must have N linearly independent rows if it is to span E^N.

Theorem 2. Given a non-recursive filter $\underline{G}$ of length N, and an N-dimensional input sequence $\{\underline{x}_j\}_0^\infty$, the weight vector $\underline{w}_j$ obtained by α-LMS adaption $(0 < \alpha < 2)$ converges to $\underline{G}$ as $j \to \infty$.

Theorem 2 is proved in Appendix A. The method of proof is illustrated in Fig. 10 for N = 2. The input vectors $\underline{x}_j$ are operated on in the convergence proof in groups of N by a transformation which yields

$$\underline{\eta}_j = \underline{x}_j - \tilde{\underline{x}}_j$$

such that $\underline{\eta}_j$ is orthogonal to all N - 1 previous $\underline{x}_j$'s in the group.

This is a Gram-Schmidt transformation which restarts every time N vectors have been processed. Thus, the quantity $\|\underline{w}_j - \underline{G}\|$ at any time j is bounded above by the convergence due to the $\underline{\eta}_j$'s. (This follows from Theorem 1 and the orthogonal nature of the $\underline{\eta}_j$'s.) The $\underline{\eta}_j$'s are the "new information" and are all non-zero due to the fact that by definition $\underline{\underline{B}}_j$, has rank N. Note that the closer $\underline{\eta}_j$ is to $\underline{x}_j$ (the closer δ is to 1) the faster $\underline{w}_j$ will converge to $\underline{G}$. For the example considered in Section 2.C, $\underline{n}_j = \underline{x}_j$ and $\delta = 1$ because the input sequence was itself orthogonal. The minimum rate of convergence is determined by the size of δ.

It is of interest to note that the assumption of an N-dimensional input sequence $\{\underline{x}_j\}_o^\infty$ is more restricted than is necessary to guarantee unbiased convergence. Assume that we do have $M - N$ (where $M > N$) dependent vectors in the sequence $\{\underline{x}_j\}_o^\infty$. By Theorem 1 we know that $\|\underline{w}_j - \underline{G}\|$ is non-increasing. Also, when a dependent vector $\underline{x}_j$ is used to adapt the weight vector $\underline{w}_j$, the upper bound defined by the $\underline{\eta}$'s (see Fig. 10) does not change because $\underline{\eta}_j$ is equal to zero. Thus, the upper bound does not change for $M - N$ adaptionsbut decreases normally for the remaining N adaptions. Therefore, the first part of the definition of an N-dimensional input sequence can be modified to read ... "such that there exists a finite $M \geq N$ and the sequence of vectors $\{\underline{x}_j\}_{j-M}^{j-1}$ has N linearly independent element vectors for all $j \geq 2M$." ... If we look at any group of M input vectors we are guaranteed that $\|\underline{w}_{j+M} - \underline{G}\| < \|\underline{w}_j - \underline{G}\|$. Since M is finite, this occurs an infinite number of times and we have

$$\lim_{j \to \infty} \underline{w}_j = \underline{G} \quad .$$

D. CONVERGENCE OF THE ADDITIVE-OUTPUT-NOISE SYSTEM

It is hard to imagine a real-world process that does not have some form of additive output noise. In addition, one inevitably introduces noise when trying to make measurements. The measurement noise can be reduced but it is never zero. For the case of additive output noise n_j we have

$$y_j = \underline{x}_j^T \underline{G} + n_j$$

and thus (2.2) is

$$\epsilon_j = \underline{x}_j^T (\underline{G} - \underline{w}_j) + n_j$$

Because n_j is a random variable it is necessary to look at the behavior of the expected value of the weight vector $\overline{\underline{w}}_j$ as $j \to \infty$. To start with let us consider the system where the noise process is zero mean:

Theorem 3. Given a non-recursive filter $\underline{G}$ of length N with additive output noise (iid, zero mean and finite variance σ^2) and an N-dimensional input sequence $\{\underline{x}_j\}_o^\infty$; then the expected value of the weight vector $\overline{\underline{w}}_j$ obtained by α-LMS adaption $(0 < \alpha < 2)$ converges to $\underline{G}$ as $j \to \infty$ and an upper bound for the mean-square error is

$$\lim_{j \to \infty} \overline{\left[\underline{x}_j^T(\underline{w}_j - \underline{G})\right]^2} \leq \frac{\alpha \sigma^2 N \|\underline{x}_{max}\|^2}{\delta(2 - \alpha\delta)\|\underline{x}_{min}\|^2}$$

Theorem 3 is proved in Appendix A.

Thus given a zero mean noise source which is uncorrelated with the input sequence $\{\underline{x}_j\}_o^\infty$, the expected value of the weight vector $\overline{\underline{w}}_j$ converges to $\underline{G}$ as $j \to \infty$. In addition we have that the MSE (mean square error) is a function of the adaption coefficient α. Thus the smaller α is the smaller the MSE will be, but at the same time the slower the rate of convergence will be. In practice, α is made close to one during periods of initialization or rapid changes in the physiological process. After these periods α is then reduced. For the experiments described in Section 4, α was made as small as 0.2. Note also that the MSE is a function of the maximum and minimum input vectors. The fact that the bound is divided by $\|\underline{x}_{min}\|^2$ is not unreasonable. Consider the example where we have a zero input and a non-zero output (due to the noise source). In this situation the MSE is not well defined.

Depending on the system, this may be considered an undesirable feature. In addition there is little reason to believe that the physiological systems one may want to model have zero mean additive output noise. To handle bias in the observation noise it is necessary to introduce a non-linear system.

<u>Definition</u>. The augmented weight vector $\underline{w}'_j$ is the vector

$$\underline{w}'_j \triangleq \begin{bmatrix} w_o \\ w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}$$

at time j, where the augmented input vector $\underline{x}'_j$ is

$$\underline{x}'_j \triangleq \begin{bmatrix} 1 \\ --- \\ \underline{x}_j \end{bmatrix}$$

The new weight w_o is referred to as the bias weight because under suitable conditions its expected value $\overline{w}_o$ converges to the mean value (bias) of the process. Note that the resulting model is non-linear,

$$y_j = \sum_{k=1}^{N} w_k x_{j-k+1} + w_o \quad .$$

Now we have the following result:

<u>Theorem 4</u>. Given a non-recursive filter $\underline{G}$ of length N with additive output noise (iid, mean β and finite variance σ^2) and an N-dimensional input sequence $\{\underline{x}_j\}_o^\infty$ then the expected value of the augmented weight vector $\overline{\underline{w}}_j'$ obtained by α-LMS adaption $(0<\alpha<2)$ converges, namely $\overline{\underline{w}}_j$ converges to $\underline{G}$ and $\overline{w}_o$ converges to β as $j \to \infty$, and an upper bound for the mean square error is

$$\lim_{j\to\infty} \overline{\left[\underline{x}_j^T(\underline{w}_j - \underline{G})\right]^2} \leq \frac{4\sigma^2 N\|\underline{x}_{max}\|^2}{\alpha\delta(2-\alpha\delta)}$$

Theorem 4 is proved in Appendix A.

The augmented α-LMS Algorithm thus has the very nice property that $\lim_{j\to\infty} \overline{w}_o = \beta$ and $\lim_{j\to\infty} \overline{\underline{w}}_j = \underline{G}$. Furthermore, we notice that the norm squared of the minimum input vector is one. Thus, for all finite input vectors $\underline{x}_j$, the MSE is finite. In the bound for the MSE, α^2 has been replaced by its upper bound of 4. The author conjectures that the MSE bound is still a function of α^2. The difficulty of working with a nonzero mean noise arises in the proof because the α-LMS Algorithm is not a linear process , and the mean β is not known <u>a priori</u>.

E. CONVERGENCE OF THE HIGH AND LOW ORDER MODELS

From both a practical and a theoretical point of view it is important to know how the α-LMS Algorithm behaves when the length of the model is not equal to the system being modeled. Experimentally it is unlikely that the length of the impulse response, $\underline{G}$, will be known exactly a priori. Theoretically it is advantageous to make N as small as possible because this reduces the MSE bound. The usual rule of thumb is to make the total delay time of the tapped delay line of the adaptive model at least as long as or longer than $\underline{G}$, in which case we have:

Corollary 1. Given a non-recursive filter $\underline{G}$ of length $M < N$ and an N-dimensional input sequence $\{\underline{x}_j\}_0^\infty$ then the weight vector $\underline{w}_j$ obtained by α-LMS adaption $(0 < \alpha < 2)$ converges to $\hat{\underline{G}}$ as $j \to \infty$ where

$$\hat{\underline{G}} = \begin{bmatrix} \underline{G} \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \left.\vphantom{\begin{matrix}0\\0\\\vdots\\0\end{matrix}}\right\} N-M \text{ zeros}$$

.

Proof of Corollary 1.

Since the addition of feedforward terms of zero gain does not affect the output of $\underline{G}$, substitute $\hat{\underline{G}}$ for $\underline{G}$ in the proof of Theorem 2. ∎

This completes the proof of Corollary 1.

Thus, when the model is of higher order than the physiological system, the primary effect is increased MSE.

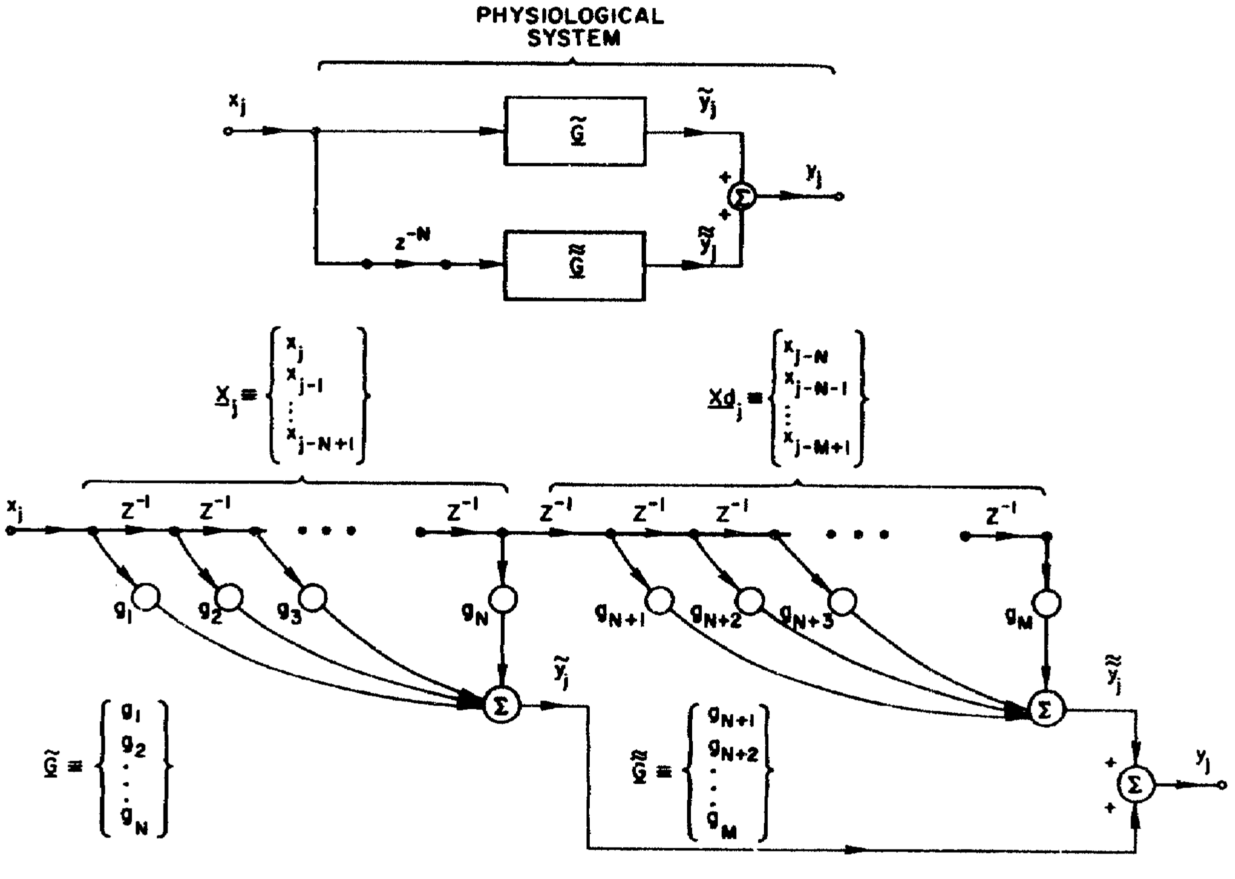

PHYSIOLOGICAL SYSTEM
x_j
$\tilde{G}$
$\tilde{y}_j$
z^{-N}
$\approx G$
$\approx y_j$
+
+
Σ
y_j
$\underline{X}_j \equiv \{x_j, x_{j-1}, \dots, x_{j-N+1}\}$
$\underline{Xd}_j \equiv \{x_{j-N}, x_{j-N-1}, \dots, x_{j-M+1}\}$
Z^{-1}
g_1
g_2
g_3
g_N
g_{N+1}
g_{N+2}
g_{N+3}
g_M
$\tilde{\underline{G}} \equiv \{g_1, g_2, \dots, g_N\}$
$\approx \underline{G} \equiv \{g_{N+1}, g_{N+2}, \dots, g_M\}$

Fig. 11. Representation of the Impulse Response in Two Parts: first part $\underline{\tilde{G}}$ and second part $\underline{\approx G}$.

In the event that the model is of lower order than the physiological system, the problem is much more complicated. Decompose the filter $\underline{G}$ into two parts $\tilde{\underline{G}}$ and $\overset{\approx}{\underline{G}}$ where $\tilde{\underline{G}}$ is the first N elements of $\underline{G}$ and $\overset{\approx}{\underline{G}}$ is the remaining elements (see Fig. 11). In this way $\tilde{y}_j = \underline{x}_j^T\tilde{\underline{G}}$ and $\overset{\approx}{y}_j = \underline{xd}_j^T\overset{\approx}{\underline{G}}$ so that (2.2) is given by

$$\epsilon_j = \underline{x}_j^T(\tilde{\underline{G}} - \underline{w}_j) + \underline{xd}_j^T\overset{\approx}{\underline{G}} \quad .$$

Now unless we can make some statements about the long term behavior of the input sequence

$$\left\{ \begin{array}{c} \underline{x}_j \\ \hline \underline{xd}_j \end{array} \right\}_o^\infty ,$$

we cannot say what $\lim_{j \to \infty} \underline{w}_j$ is equal to. To see this we make the following construction. Suppose we interpret $\overset{\approx}{y}_j$ as being an additive output noise, then $\overset{\approx}{y}_j$ must have a stationary mean and variance if Theorem 4 is to apply. While this may be the case for some systems, this is certainly not required in the definition of an N-dimensional input sequence. But if the input sequence can be modeled by a stochastic process and if $\underline{x}_j$ and $\underline{xd}_j$ are uncorrelated, then Theorem 4 applies.

This being the case, it is advisable to design the identification system to be as long as if not longer than the maximum possible length of the impulse response of the physiological system.

3. FORWARD TIME CONTROLLER

A. THE MINIMIZATION PROBLEM

The problems in optimal control are typically associated with dynamic systems evolving in time where the systems are described by a set of differential equations [36-39]. Mathematically formulated, the determination of the extrema (optimal points) of these functionals is a problem in the calculus of variations. However, in the case of the adaptive model controller of Fig. 7, the dynamic system is modeled by

$$y_j = \underline{x}_j^T \underline{w}_j \quad . \tag{3.0}$$

This mathematical formulation (parameterization) of the dynamic system in conjunction with the assumption that the reference input r_i is known for finite future time leads to a function minimization algorithm which does not use the calculus of variations. The development of this algorithm is the thesis of this section.

Let L = integer constant

N = the number of weights

j = the current time, and

$T = LN + j$

The reference input r_i and a set of constraints a_i, b_i, c_i & d_i are assumed to be known for all $i = j+1, j+2, \ldots T$. Thus the problem is to determine x_i for all $i = j+1, j+2, \ldots T$ such that

$$\sum_{i=j+1}^{T} (y_i - r_i)^2 \tag{3.1}$$

is minimized subject to the constraints

$$a_i \leq y_i \leq b_i$$

$$c_i \leq x_i \leq d_i$$

where

$$y_i = \sum_{k=1}^{N} w_k x_{i-k+1} \qquad i = j+1, j+2, \ldots T$$

The blood pressure (output)constraints a_i and b_i are needed because there is a minimum value above which the blood pressure must remain at all times (note that $a_i < 0$ has no meaning in this problem!) and likewise, there is a maximum value of blood pressure that is safe for the patient. The drug rate (input) constraints c_i and d_i correspond to the fact that there is a minimum and maximum drug flow rate. The minimum flow rate is that rate which will keep a blood clot from forming in the catheter. If a blood clot forms, the catheter must be disconnected from the drug supply and flushed to clear the blood clot. The maximum flow rate is determined by the maximum amount of fluid that can be added to the blood without causing undesired side effects.

The minimization of the squared control error (3.1) is used for two reasons. First, it produces very satisfactory control, and second, it allows a considerable reduction in the mathematical complexity of the forward time calculations needed to achieve closed loop control. The forward time control algorithm described here is thus optimal in a minimum squared error sense with respect to "controlling" the discrete non-recursive filter model. The optimality of this approach carries over to the entire control system when the model has converged.

Before we begin the solution of the minimization problem, it is instructive to discuss some of the overall system requirements and model properties which have been used in choosing the most desirable algorithm. Notice that this representation of the physiological system (the "plant" to be controlled) gives us not only a simple formula for determining the output given the input, but it also gives us the state* of the model. This is very important in achieving control because we are dealing with a feedback system and must be able to control at all times. That is, we do not want to wait until the transients from the last correction have died out before we make the next correction.

The overall system requirements arise from the need to run this model reference system in real-time on a mini-computer. These requirements are:

1. The algorithm should make efficient use of memory since the total amount of memory available for this part of the system is small (3-4K words).
2. The number of arithmetic operations should be minimized because these operations are not very fast on mini-computers and there is a limited amount of time available.
3. The algorithm should calculate the solution iteratively. An algorithm which obtains an exact solution in τ seconds is useless if less than τ seconds is available, while an algorithm which iteratively refines the initial guess has an improved "solution" at all times.

*The model is a non-recursive digital filter; therefore the state of the model is precisely $\underline{x}_j$.

4. The algorithm should converge rapidly and in a finite number of steps even when many constraints are active.

B. THE FORWARD TIME CALCULATION

The solution of (3.1) is divided into four steps (computation phases). In this way the "solution" of (3.1) is refined in proportion to the amount of processing time available at time j. Physiological systems quite typically have some amount of transport delay. This means that several of the first weights along the non-recursive digital filter model would be equal to zero. In practice these weights are not exactly equal to zero, but instead vary by some small amount ξ about zero (see Table 1). This is handled experimentally by labeling all $|w_i| \leq \xi$ as being zero. The first nonzero weight $w_{\mathcal{J}}$ is then referred to as the first weight of $\underline{w}_j$ where $\underline{w}_j$ now has $N-\mathcal{J}$ elements. At the same time, the model is iterated forward $\mathcal{J}$ cycles. In this way the model is transformed into an equivalent system with no delay, and the new w_1 is not equal to zero (this is needed because we will have to divide by w_1 to get the solution to (3.1)).

The first numerical task is to get a feasible* solution as fast as possible (recall that the amount of time available for the solution of (3.1) is <u>not</u> known): This is done by recognizing that if the plant is stable then it will have a steady state output which is equal to r_i. We can then use (3.0) to find a feasible input. Thus, we have

Phase 1:

Find the steady state solution x_{ss} and then let $x_i = x_{ss}$ for all $i = j+1, j+2, \ldots T$. Thus we have

*A solution is feasible if the constants are satisfied. Thus x_i is feasible iff $c_i \leq x_i \leq d_i$ and $a_i \leq y_i \leq b_i$.

$$x_{ss} = \frac{\frac{1}{LN} \sum_{i=j+1}^{T} r_i}{\sum_{k=1}^{N} w_k}$$

This solution is interior to the boundary of the feasible domain, but it is not likely to be the optimal solution.

Phase 2:

Check to see if the problem can be solved with zero error. Set $(y_i - r_i)^2 = 0$ for all $i = j+1, j+2, \ldots T$ and solve for x_i:

$$x_i = \begin{cases} \hat{x}_i, & c_i \leq \hat{x}_i \leq d_i \\ \text{undefined otherwise} \end{cases} \qquad (3.2)$$

where

$$\hat{x}_i = \frac{r_i - \sum_{k=2}^{N} w_k x_{i-k+1}}{w_1} \qquad (3.3)$$

If x_i is undefined for any i then Phase 2 fails and we proceed with Phase 3; otherwise we have the optimal solution for (3.1) and are done.

Phase 3 is a heuristic procedure motivated by the solutions obtained using Rosen's gradient projection method (Phase 4). It was found that the solutions for the optimal drug rates obtained in Phase 4 could be fitted with exponentially decaying envelopes. Intuitively this is reasonable because we would expect that after a change in the reference input, the drug rate would change greatly at first and then settle to a steady state value. This heuristic procedure we call the "clipped" solution because the drug rates x_i are not allowed to be outside of this envelope. Thus, the heuristic procedure is that if the drug rate $\hat{x}_i$ calculated in (3.3) is outside of the envelope, it is replaced by the value of the envelope (see (3.4)). (Note that the results of the Phase 2 calculation are directly usable in Phase 3.)

Phase 3:

Check to see if the "clipped" solution will satisfy the output (blood pressure) constraints. Solve for x_i, $i = j+1, j+2, \ldots T$ until

$$y_i < a_i \quad \text{or} \quad y_i > b_i$$

where

$$x_i = \begin{cases} UB_i, & \hat{x}_i \geq UB_i \\ \hat{x}_i, & LB_i < \hat{x}_i < UB_i \\ LB_i, & \hat{x}_i \leq LB_i \end{cases} \tag{3.4}$$

and where

$$\hat{x}_i = \frac{r_i - \sum_{k=2}^{N} w_k x_{i-k+1}}{w_1}$$

$$UB_i = \min[d_i, UC_i]$$

$$LB_i = \max[c_i, LC_i]$$

$$UC_i = x_{ss} + \Upsilon \exp[-\Psi(i-j)]$$

$$LC_i = x_{ss} - \Upsilon \exp[-\Psi(i-j)]$$

where B is defined by

$$\Psi = \frac{1}{NL} \ln(\Upsilon/\zeta)$$

where

$$\Upsilon \triangleq \max\left|x_{ss} - x_i\right| \quad \text{allowed at time } i = j$$

and

$$\zeta \triangleq \max \left| x_{ss} - x_i \right| \quad \text{allowed at time } i - T$$

If Phase 3 is successful use its results as the initial guess in Phase 4; else use the results from Phase 1. In order to describe the method used in Phase 4 it is necessary to formulate (3.1) as a nonlinear programming problem. Recall that

$$y_i = \underline{x}_i^T \underline{w}_i \qquad i = j+1, j+2, \ \ldots \ T$$

or in matrix form

$$\begin{bmatrix} y_{j+1} \\ y_{j+2} \\ \vdots \\ y_{j+N} \\ \hdashline y_{j+N+1} \\ \vdots \\ y_T \end{bmatrix} = \left[\begin{array}{c:c} \underbrace{\begin{matrix} w_N \cdots & & w_2 \\ & w_N \cdots & w_3 \\ 0 & & \ddots \\ & & \end{matrix}}_{\substack{N \\ \text{CONSTANT}}} & \begin{matrix} w_1 & & & 0 \\ w_2 & w_1 & & \\ w_N \cdots & w_2 & w_1 & \end{matrix} \\ \hdashline 0 & \begin{matrix} \ddots & & & 0 \\ 0 & & \ddots & \\ & w_N \cdots & w_2 & w_1 \end{matrix} \end{array}\right] \begin{bmatrix} \left.\begin{matrix} x_{j-N+2} \\ x_{j-N+3} \\ \vdots \\ x_j \end{matrix}\right\} \text{known} \\ \hdashline \left.\begin{matrix} x_{j+1} \\ x_{j+2} \\ \vdots \\ x_T \end{matrix}\right\} \substack{\text{to be} \\ \text{optimized}} \end{bmatrix}$$

where the past inputs $x_{j-N+2} \ldots x_j$ are known and the future inputs $x_{j+1} \ldots x_T$ are to be optimized. Notice that the upper left hand block of this matrix can be replaced by a constant vector $\underline{h}_i$, say; thus we have

$$\begin{bmatrix} y_{j+1} \\ y_{j+2} \\ \vdots \\ y_{j+N} \\ y_{j+N+1} \\ \vdots \\ y_T \end{bmatrix} = \begin{bmatrix} h_{j+1} \\ h_{j+2} \\ \vdots \\ h_{j+N} \\ h_{j+N+1} \\ \\ h_T \end{bmatrix} + \begin{bmatrix} w_1 & & & & & 0 \\ w_2 & w_1 & & & & \\ \vdots & & \ddots & & & \\ w_N \cdots & w_2 & w_1 & \ddots & & \\ & \ddots & & & \ddots & \\ 0 & & w_N \cdots & & w_2 & w_1 \end{bmatrix} \begin{bmatrix} x_{j+1} \\ x_{j+2} \\ \vdots \\ \\ \\ x_T \end{bmatrix} \qquad (3.5)$$

or letting $i = j + 1$ for convenience, and defining the vectors, we write (3.5) as

$$\underset{\sim}{y}_i = \underset{\sim}{h}_i + \underline{\underline{\mathcal{W}}}\,\underset{\sim}{x}_i$$

where

$$h_\kappa = \begin{cases} \displaystyle\sum_{k=\kappa-j}^{N} w_{k+1}\, x_{j-k+1} & , \quad \kappa < j + N \\ 0 & , \quad \kappa \geq j + N \end{cases} \tag{3.6a}$$

and

$$\mathcal{W}_{\kappa m} = \begin{cases} 0 & \text{for} \quad m > \kappa \\ 0 & \text{for} \quad \kappa > N \quad \text{and} \quad m \leq \kappa - N \\ w_{\kappa+1-m} & \text{elsewhere} \\ \text{(at time } j) & \end{cases} \tag{3.6b}$$

Note that $\underline{\underline{\mathcal{W}}}$ is in lower triangular band form with bandwidth N, and that all of the elements on any given diagonal are identical. Therefore using (3.6b), only N computer words are required to store $\underline{\underline{\mathcal{W}}}$ instead of $(LN)^2$ words. Furthermore, $\underline{\underline{\mathcal{W}}}$ inverse exists because w_1 is not equal to zero and is easily obtained by one front-solve* operation. We can now construct the following quadratic minimization problem. Minimize

$$\sum_{i=j+1}^{T} (y_i - r_i)^2$$

where

$$y_i = h_i + \sum_{k=1}^{T-j} \mathcal{W}_{i-j,k}\, x_{j+k}$$

*When the matrix of coefficients for a system of equations is lower triangular, the solution is readily found by solving the equations one at a time from the top down; hence the name front-sol'e.

and

$$r_i \equiv \text{reference input}$$

$$a_i \leq y_i \leq b_i$$

$$c_i \leq x_i \leq d_i$$

Define the error vector as

$$\underline{z}_i \triangleq \underline{y}_i - \underline{r}_i$$

$$= \underline{\underline{\mathcal{W}}}\,\underline{x}_i + \underline{h}_i - \underline{r}_i \qquad (3.7)$$

Thus, the parameters to be optimized are

$$\underline{x}_i = \underline{\underline{\mathcal{W}}}^{-1}(\underline{r}_i - \underline{h}_i) + \underline{\underline{\mathcal{W}}}^{-1}\underline{z}_i$$

and the function we want to minimize is

$$f(\underline{z}_i) = \frac{1}{2}(\underline{z}_i^T\underline{z}_i) \qquad (3.8)$$

where the constraints are now given by

$$\underline{a}_i - \underline{r}_i \leq \underline{z}_i \leq \underline{b}_i - \underline{r}_i \qquad (3.9a)$$

and

$$\underline{c}_i - \underline{\underline{\mathcal{W}}}^{-1}(\underline{r}_i - \underline{h}_i) \leq \underline{\underline{\mathcal{W}}}^{-1}\underline{z}_i \leq \underline{d}_i - \underline{\underline{\mathcal{W}}}^{-1}(\underline{r}_i - \underline{h}_i) \qquad (3.9b)$$

The desired form for the constraint equation in the nonlinear programming problem is

$$\underline{n}_k^T\underline{z}_i \geq \ell_k \qquad k = 1,2,\ \ldots\ 4LN$$

where

$$\underline{n}_k^T \underline{n}_k = 1 \qquad k = 1,2, \ldots 4LN$$

Rearranging the constraint equations (3.9a) and(3.9b) into this form we get(without normalizing)

$$\left[\begin{array}{c} \underline{\underline{I}} \\ \hdashline -\underline{\underline{I}} \\ \hdashline \underline{\underline{\mathcal{W}}}^{-1} \\ \hdashline -\underline{\underline{\mathcal{W}}}^{-1} \end{array}\right] \left[\underline{z}_i\right] \geq \left[\begin{array}{c} \underline{a}_i - \underline{r}_i \\ \hdashline -\underline{b}_i + \underline{r}_i \\ \hdashline \underline{c}_i - \underline{\underline{\mathcal{W}}}^{-1}(\underline{r}_i - \underline{h}_i) \\ \hdashline -\underline{d}_i + \underline{\underline{\mathcal{W}}}^{-1}(\underline{r}_i - \underline{h}_i) \end{array}\right] \qquad (3.10)$$

The motivation for this formulation is now apparent because we have that the gradient of f is given by

$$\underline{g} = \mathrm{grad}(f(\underline{z}_i)) = \underline{z}_i$$

and the Hessian matrix of second partial derivatives of f is given by

$$\underline{\underline{G}}(\underline{z}_i) = I \quad . \qquad (3.11)$$

One can now apply Rosen's gradient projection method. In general, this method gives linear convergence since the direction of steepest descent is taken. However, in our case, where the Hessian is I, the convergence is quadratic, and hence the minimum of $f(\underline{z}_i)$ in any given subspace can be found in one step. A general description of Rosen's algorithm for the special case where $\underline{\underline{G}}$ is the identity matrix, is given in Appendix B. A programmed version of Phase 4 is given in Appendix C.

Phase 4:

1. Using (3.7) we set

$$\underline{z}^{o} = \underline{\underline{W}}\,\underline{x}_i + \underline{h}_i - \underline{r}_i$$

and

$$\underline{s} = -\underline{z}^{o}$$

where $\underline{z}^{k}$ denotes the error vector at the k^{th} iteration in solving for the optimal $\underline{z}_i$, and $\underline{h}_i$ is given by (3.6a). It is known from previous calculations in Phases 1, 2, & 3, that $\underline{z}^{o}$ is feasible (all constraints are satisfied) thus, the number of active constraints q is initially equal to zero. Also compute $\underline{\underline{W}}^{-1}$ and $\underline{\ell}$ which is equal to the right hand side of (3.10). As a matter of notation, we will let

$$\underline{\underline{N}}_q = \{\underline{n}_1, \underline{n}_2, \ldots, \underline{n}_q\}$$

and $\underline{\underline{P}}_q = \underline{\underline{Q}}\,\underline{\underline{N}}_q$ where $\underline{\underline{Q}}^T\underline{\underline{Q}} = \underline{I}$ and $\underline{\underline{P}}_q$ is upper triangular.

2. Determine

$$\lambda^{*} = \min_{\lambda_{jn} \geq 0} = \left[\lambda_{jn} = \frac{\ell_{jn} - \underline{n}_{jn}^{T}\underline{z}^{k}}{\underline{n}_{jn}^{T}\underline{s}} \;\middle|\; jn \notin \begin{Bmatrix}\text{active}\\ \text{constraints}\end{Bmatrix} \right]$$

3. If $\lambda^{*} \geq 1$ then go to step 6, else add the corresponding column $\underline{n}_{jn}$ to $\underline{\underline{N}}_q$ and form

$$\underline{\underline{P}}_{q+1} = \left[\begin{array}{c|c} \underline{\underline{P}}_q & \underline{\gamma} \\ \hline 0 & p \end{array}\right]$$

where

$$\underline{\underline{P}}_q^T \underline{\gamma} = \underline{\underline{N}}_q^T \underline{n}_{jn}$$

and

$$p = (1 - \underline{\gamma}^T \underline{\gamma})^{\frac{1}{2}}$$

Set

$$q = q + 1$$

and

$$\underline{z}^{k+1} = \underline{z}^k + \lambda^* \underline{s} \quad .$$

4. Compute the vector $\underline{\alpha}$ where

$$\underline{\underline{P}}_q^T \underline{\beta} = \underline{\ell}_q$$

and

$$\underline{\underline{P}}_q \underline{\alpha} = \underline{\beta}$$

Let

$$\alpha^* = \min \left[\alpha_j \,\middle|\, j \in \left\{ \begin{matrix} \text{active} \\ \text{constraints} \end{matrix} \right\} \right]$$

5. If $\alpha^* \geq 0$, then let $\underline{s} = -\underline{z}^k + \underline{\underline{N}}_q \underline{\alpha}$ and go to step 2, else delete the corresponding column from $\underline{\underline{N}}_q$, update $\underline{\underline{P}}_q$, set $q = q - 1$ and go to step 4.

6. The optimal input is now given by

$$x_i = \frac{1}{w_1} \left[z_1^k + s_1 + r_1 - h_1 \right]$$

If $\underline{x}_i$ is desired it is given by

$$\underline{x}_i = \underline{\underline{W}}^{-1} (\underline{z}^k + \underline{s} + \underline{r} - \underline{h})$$

This completes Phase 4.

By making use of the special forms that some of the vectors and matrices have, we have that the storage requirements are as follows:

QUANTITY	MEMORY LOCATIONS
$\underline{\underline{\mathcal{W}}}$	N
$\underline{\underline{\mathcal{W}}}^{-1}$	2LN
$\underline{\underline{N}}_q$	LN
$\underline{\underline{P}}$	$\frac{LN(LN+1)}{2}$
$\underline{\gamma}$	NONE
$\underline{n}_{jn}$	NONE

4. EXPERIMENTAL RESULTS

A. OPEN LOOP RESPONSE

The first step in implementing an adaptive model controller is to obtain open loop response data. This is necessary to find out if the physiological system satisfies

$$y(t) = F(x,y,t) + n(t)$$

where $F(\cdot)$ defines a linear deterministic system and $n(\cdot)$ represents a noise source. If $F(\cdot)$ exists, then the adaptive model controller can be used. We determine N and Δ (see Section 2.B) by measuring the open loop step response and then solving for the impulse response. In our experiments on dogs this is accomplished by the use of a vasodepressor agent, Arfonad (trimethaphan camsylate) which produces a controlled state of hypotension (shock). Arfonad is primarily a ganglionic blocking agent which has the effect of opening the dog's blood pressure control loop. When the dog is in this condition, the effect of a vasopressor agent such as Levophed (l-norepinephrine) is easily measured. Figure 12 shows the effect of a step change in Levophed on the average blood pressure. The two step responses in Fig. 12 were taken sequentially on the same dog with the time between steps being about twenty minutes. Using this data it was decided that we should set $N = 20$ and $\Delta = 5$ seconds. These values for N and Δ have been satisfactory for all the dogs that we have run on this system.

Since we know that $\bar{\underline{w}}_j$ will converge to the impulse response $\underline{G}$, we would like to know what $\underline{G}$ would give these step responses. Let $I_1(t)$ be the input and $O_1(t)$ be the output; then if the input step

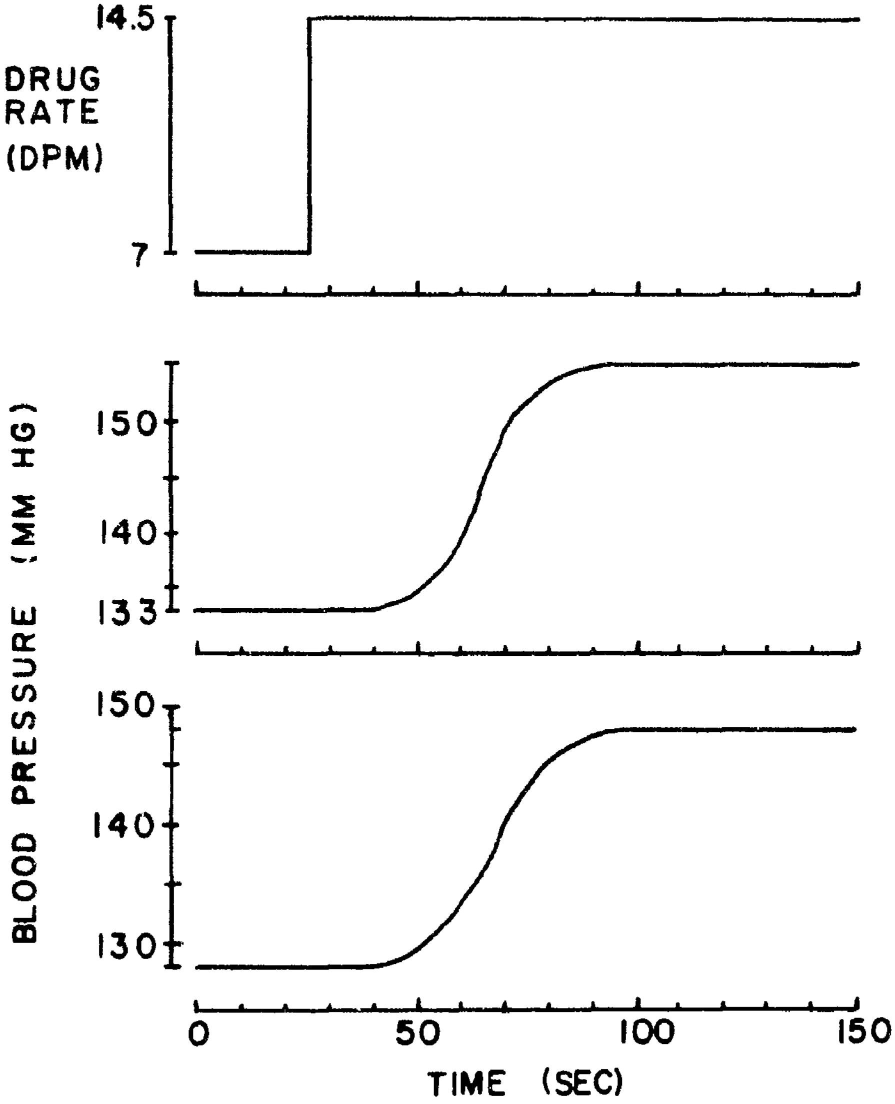


Fig. 12. Step Responses to Levophed.

was from A to B, the normalized input $I_2(t)$ is

$$I_2(t) = \frac{I_1(t) - A}{B - A} \quad .$$

If the output changed from C to D then the normalized output is

$$O_2(t) = \frac{O_1(t) - C}{B - A} \quad .$$

By looking at the data only at the times Δ, 2Δ, 3Δ, ... , we have that the elements of $\underline{G}$ are given by the iterative equation

$$g_1 = O_2(\Delta)$$

and

$$g_i = O_2(i\Delta) - O_2((i-1)\Delta) \qquad i = 2,3, \ldots, N \qquad (4.1)$$

The result of applying (4.1) to the step-response data shown in Fig. 12 is plotted in Fig. 13. A dominant feature of this data is the transport delay which is about fifteen seconds. This $\underline{G}$ or the weight vector from the previous experiment is used as the initial weight vector $\underline{W}$ when starting a new experiment. In practice, the identification process is run open loop until a good model has been formed. When the MSE is small, the loop can then be closed for automatic control.

B. CLOSED-LOOP RESPONSE

The instantaneous blood pressure was sampled every 100 milliseconds and then averaged over a 30-second period. The adaption and control cycles occurred every 5 seconds. The first 30 to 45 minutes of the experiment were typically used to calibrate the data link and to perform open loop control to verify that everything was functioning properly.

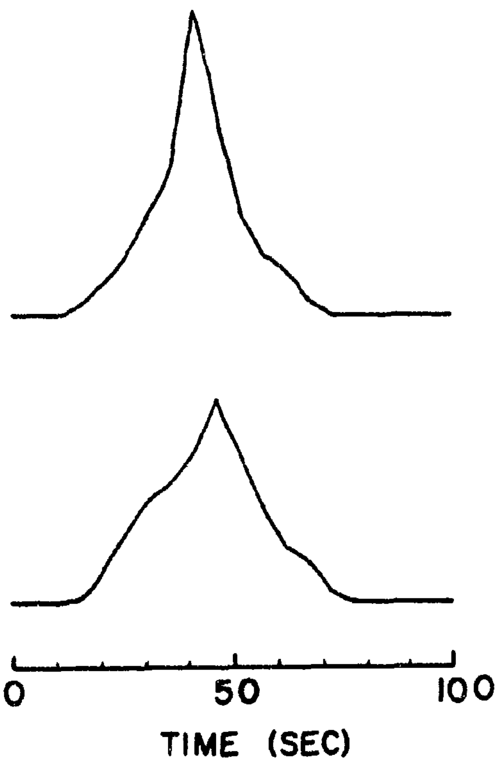


Fig. 13. The Impulse Responses Derived from the Step Response Data.

The evolution of the weight vector with time is shown in Fig. 14 (the normalized weight values are tabulated in Table 1). The successive weight vectors are plotted off to the right to aid in visualizing the sequence of weight vectors as a continuous three-dimensional surface. Note the similarity between this data and the data for $\underline{G}$ plotted in Fig. 13. The times at which these weight vectors were recorded are indicated in Fig. 15a-g as "+" marks on the line labeled $\overline{(\text{control error})^2}$.

The input (drug rate) and the output (average blood pressure) are shown in Fig. 15a-g. In addition, the model output at time $j+1$ (one step into the future) is plotted as a cross-hatched line. The line

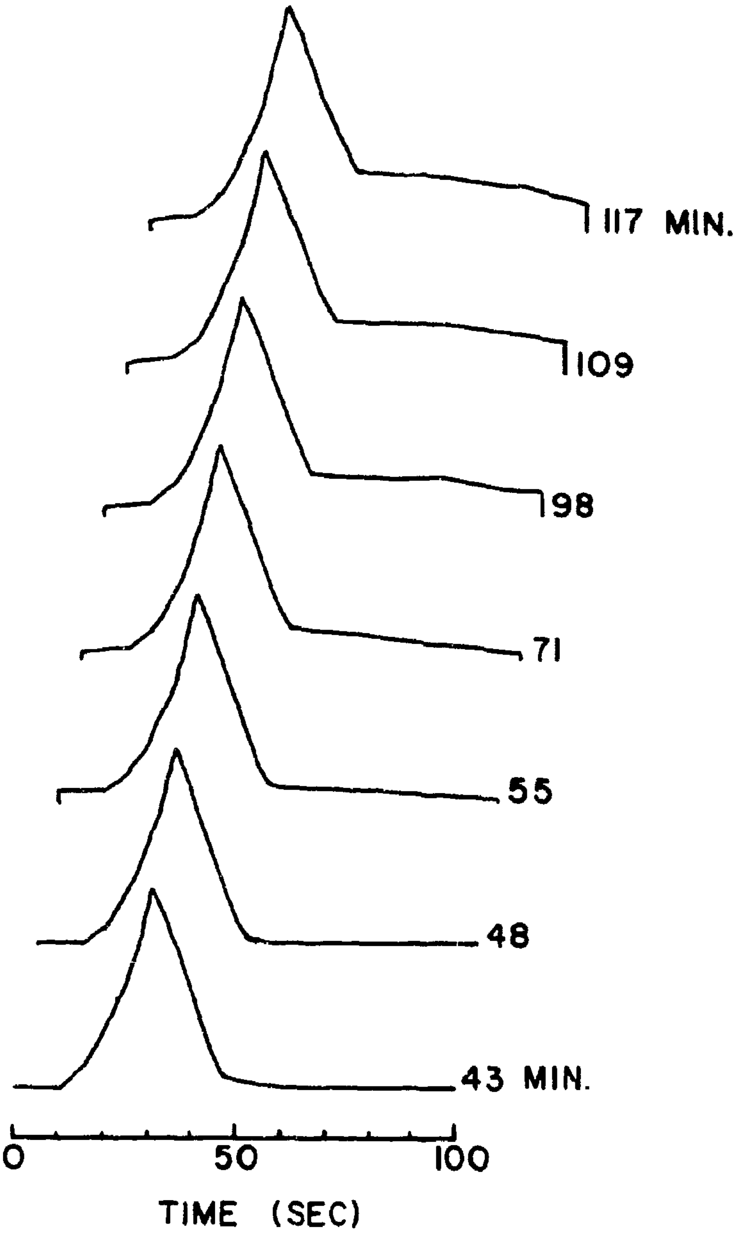


Fig. 14. Weight Values as a Function of Time During the Experiment.

NORMALIZED WEIGHT VALUES

	ELAPSED TIME (MIN)						
WEIGHT #	43	48	55	71	98	109	117
1	.0014	.0100	.0583	.0413	.0427	.0384	.0398
2	.0014	.0100	.0597	.0484	.0569	.0612	.0569
3	.0014	.0100	.0597	.0541	.0654	.0725	.0711
4	.1110	.0896	.1408	.1394	.1522	.1579	.1579
5	.3001	.2788	.3300	.3343	.3499	.3585	.3599
6	.5690	.5477	.5989	.6117	.6330	.6472	.6515
7	1.0000	.9787	1.0327	1.0555	1.0839	1.1024	1.1110
8	.6771	.6572	.7127	.7511	.7923	.8208	.8378
9	.3215	.3030	.3599	.4111	.4609	.5007	.5263
10	.0526	.0341	.0939	.1565	.2105	.2589	.2888
11	.0256	.0100	.0683	.1394	.1977	.2504	.2817
12	.0057	.0100	.0654	.1323	.1991	.2518	.2802
13	.0014	.0100	.0626	.1223	.1963	.2447	.2660
14	.0014	.0100	.0583	.1024	.1906	.2418	.2589
15	.0014	.0100	.0526	.0939	.1863	.2376	.2475
16	.0014	.0100	.0469	.0725	.1778	.2191	.2304
17	.0014	.0100	.0370	.0612	.1565	.2091	.2219
18	.0014	.0085	.0299	.0526	.1422	.1878	.1906
19	.0014	.0085	.0242	.0299	.1351	.1778	.1650
20	.0014	.0085	.0142	.0284	.1223	.1522	.1309

Table I. NORMALIZED WEIGHT VALUES AS A FUNCTION OF TIME DURING THE EXPERIMENT.

labeled $\overline{\text{(control error)}^2}$ is equal to the average value of $(y_j - r_j)^2$ where the reference input r_j (pressure set point) is indicated by a dashed line.

In Fig. 15a & b one can see not only that the system regulates the dog's blood pressure, but also the effects of learning due to the step changes in the reference input. Now, in order to dramatize the inherent stability of the system due to the continuous system identification that is occurring, several transients have been introduced. In Fig. 15c & d the effect of injecting sodium pentobarbital (a general anesthetic) is shown. This can be considered a typical situation because patients often require drugs in addition to the ones used by the control system. In Fig. 15e & f the dog's blood pressure regulation is again modified by the administration of the gasses Halothane (bromochlorotrifluoroethane) and then Amyl Nitrite (isoamyl nitrite). Halothane is an inhalation anesthetic and Amyl Nitrite is a coronary vasodilator. The reason for doing this is to demonstrate the speed at which the system can compensate for drastic short-term changes. A very similar situation occurs in the normal course of events when the present bottle of Levophed is running low and must be replaced by another. The time during which no drug is available can be minimized, but there is usually a detectable difference in concentration of Levophed between the two bottles. Fig. 15g shows how well this system regulates when the physiological system is not undergoing drastic changes.

While the analytical problems introduced by a time-varying and/or nonlinear system are still unanswered, the empirical results shown in Fig. 15a-g are a strong motivation to continue with this approach for regulating physiological systems.

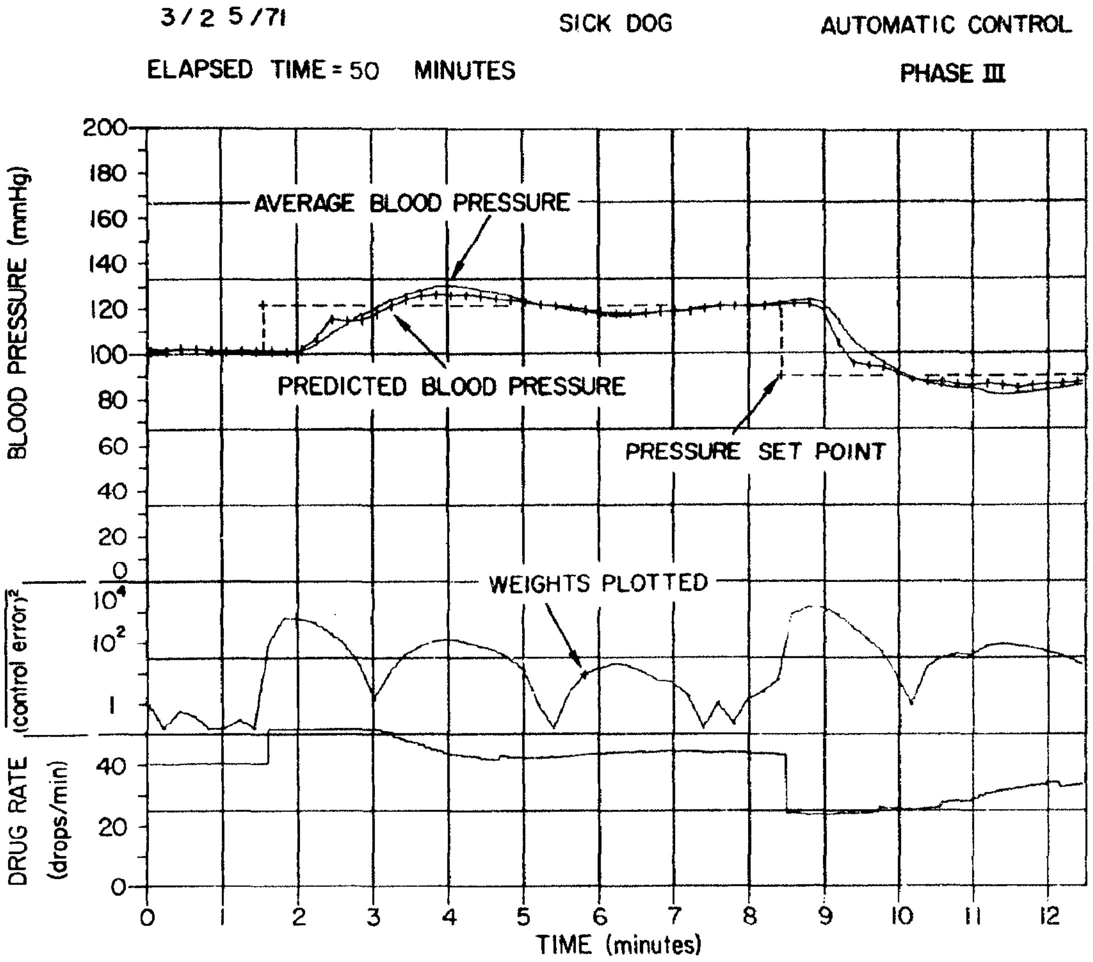


Fig. 15a. The X-Y Recorder Output Generated During the Experiment.

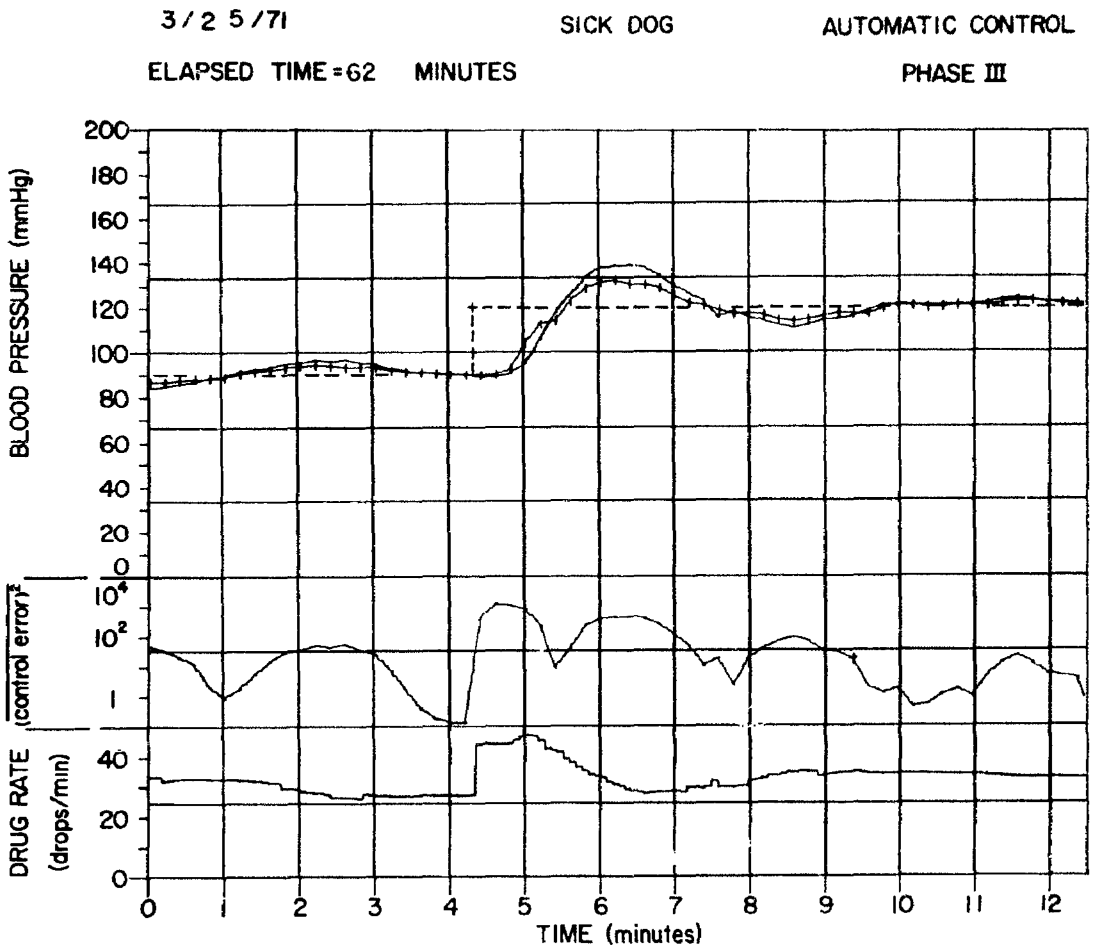


Fig. 15b. The X-Y Recorder Output Generated During the Experiment.

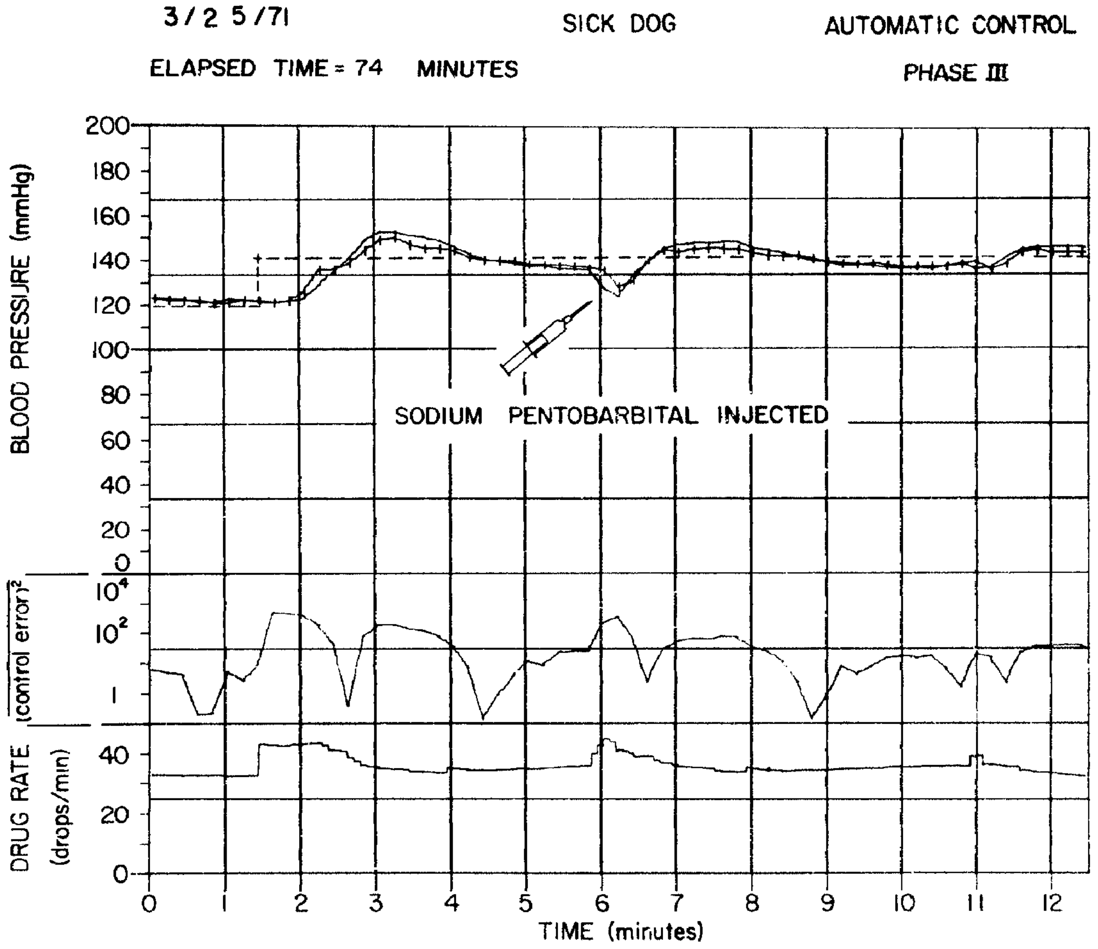
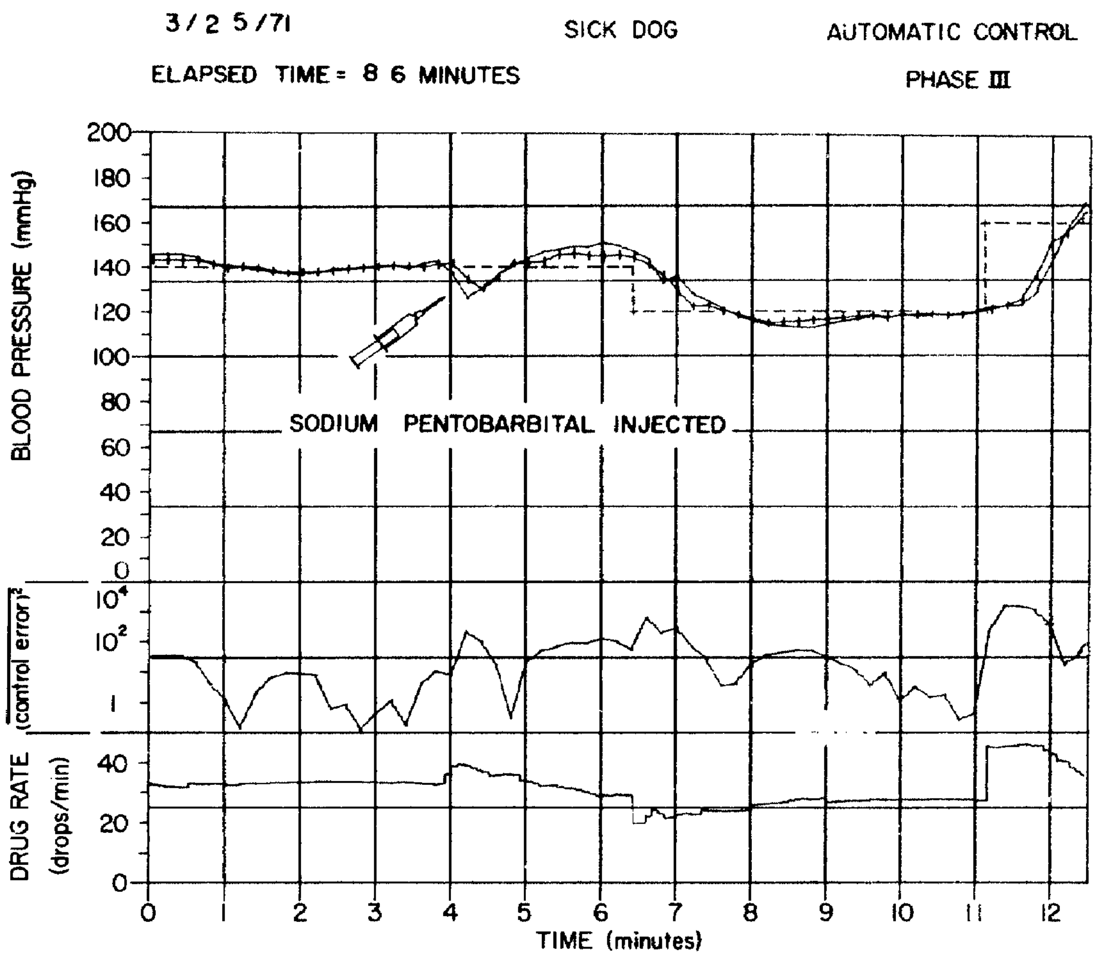


Fig. 15c. The X-Y Pecorder Output Generated During the Experiment.



Fir. 15d. The X-Y Recorder Output Generated During the Experiment.

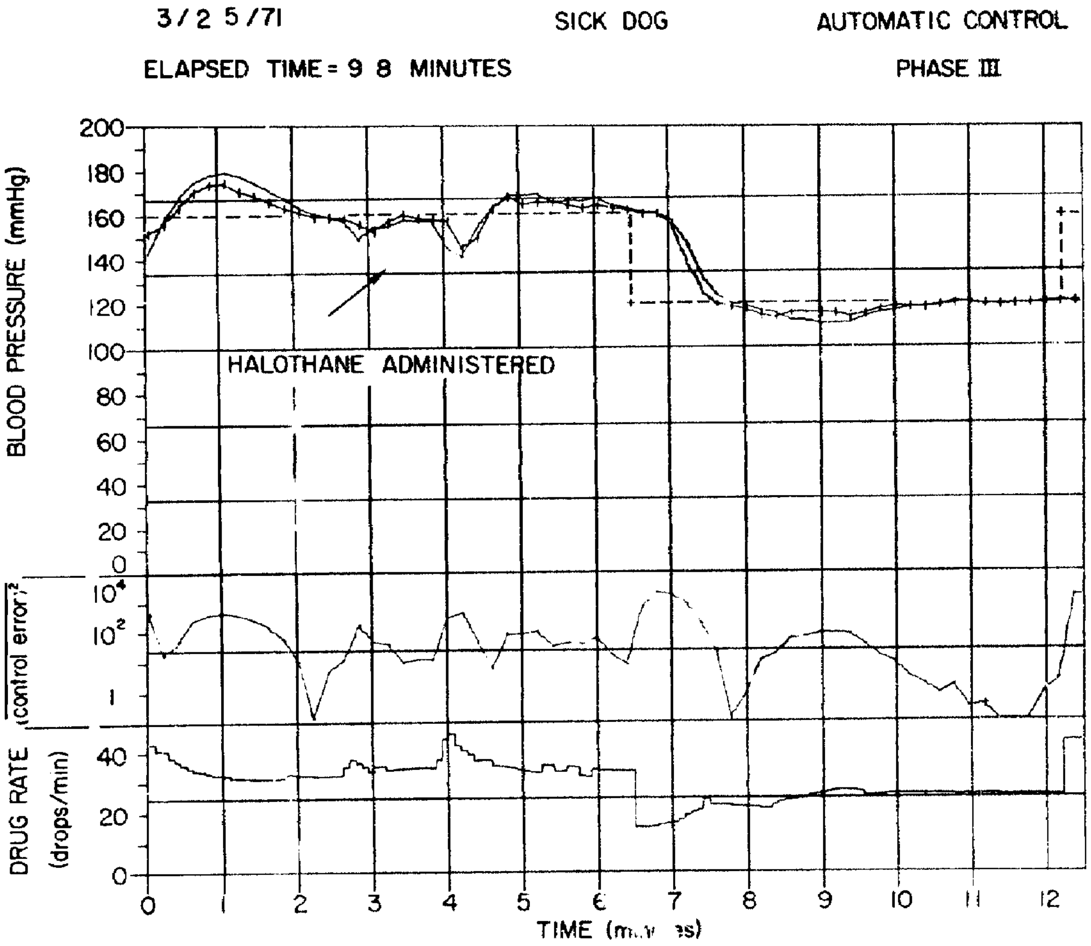


Fig. 15e. The X-Y Recorder Output Generated During the Experiment.

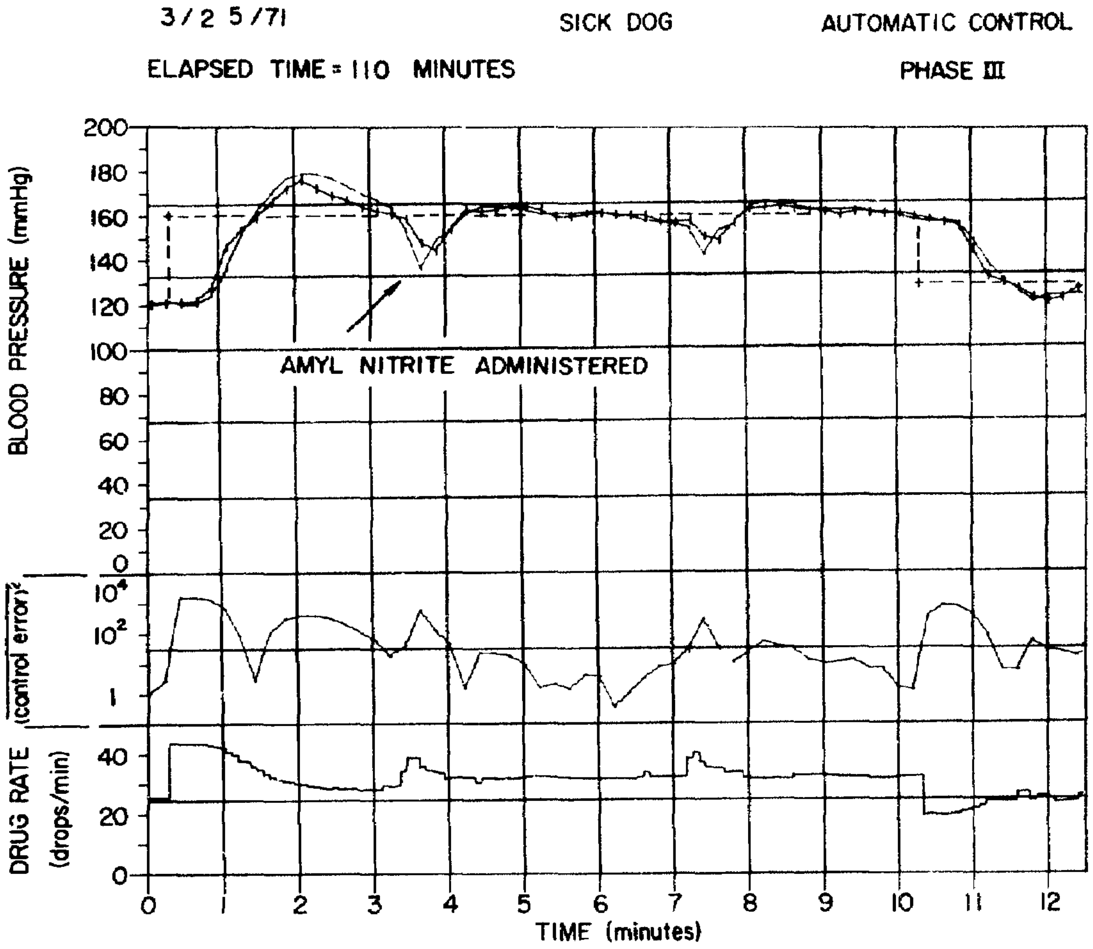


Fig. 15f. The X-Y Recorder Output Generated During the Experiment.

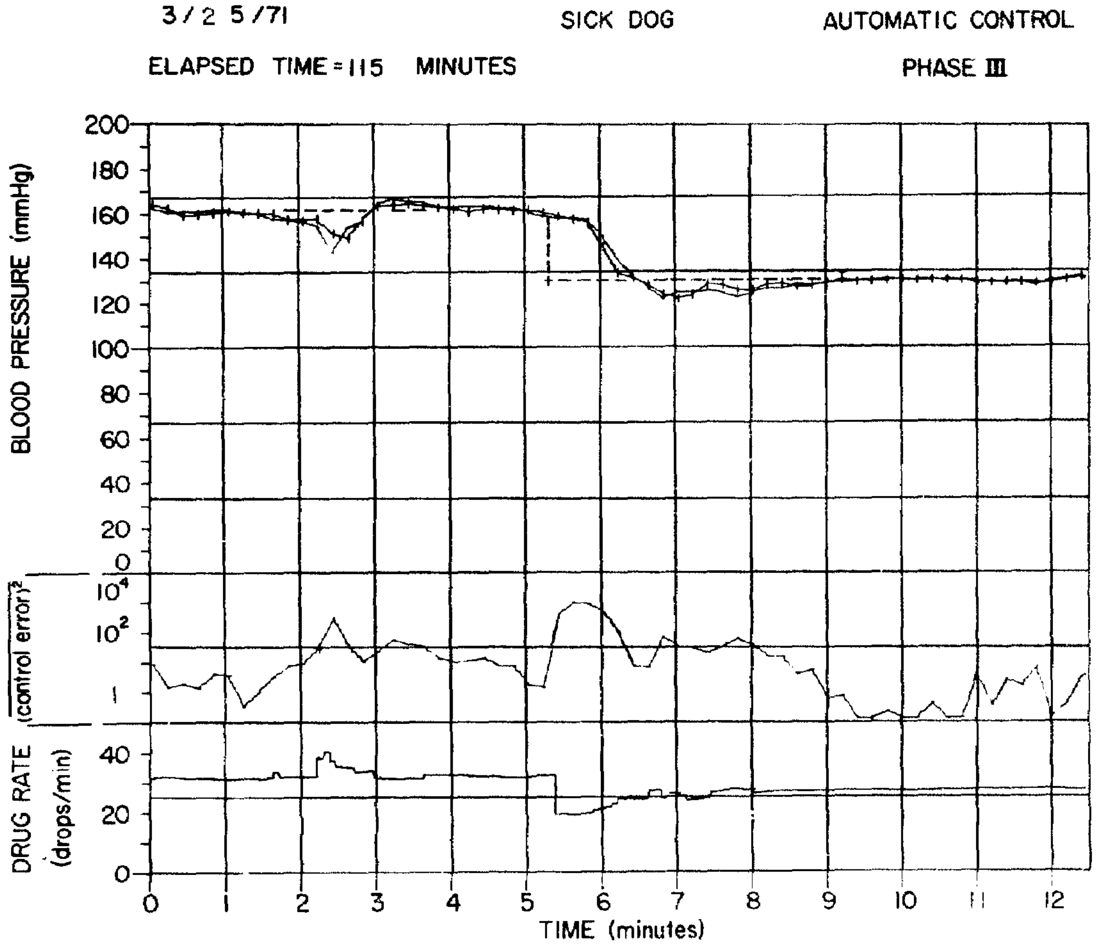


Fig. 15g. The X-Y Recorder Output Generated During the Experiment.

5. CONCLUSIONS

A. SUMMARY OF RESULTS

An original system (the adaptive model controller) for realizing an automatic therapeutic control system has been presented. It was found that this controller can be realized with a mini-computer, and that the resulting control is highly satisfactory from a medical point of view [49].

The convergence of the α-LMS Algorithm has been analyzed for deterministic inputs. Necessary and sufficient conditions for unbiased convergence were given. These conditions were found to be easy to implement and to cause little or no deterioration in the control problem. In addition, the behavior of the adaptive model where the system to be modeled is of higher or lower order, was presented.

The forward-time controller which gives minimum-squared-error with respect to the model was described. The minimization was done by a nonlinear programming technique based on Rosen's gradient projection method. The solution to this part of the forward-time calculation was optimized so that the amount of memory and computer time required was at a minimum.

An actual experiment using this system was included and described. The results of this and similar experiments have been very successful.

B. RECOMMENDATIONS FOR FURTHER WORK

The use of an adaptive multi-model controller (see Fig. 16) for blood pressure regulation would be very useful. The additional adaptive processes present no further complication in the identification part of the system. However, the optimal controller would have to be modified

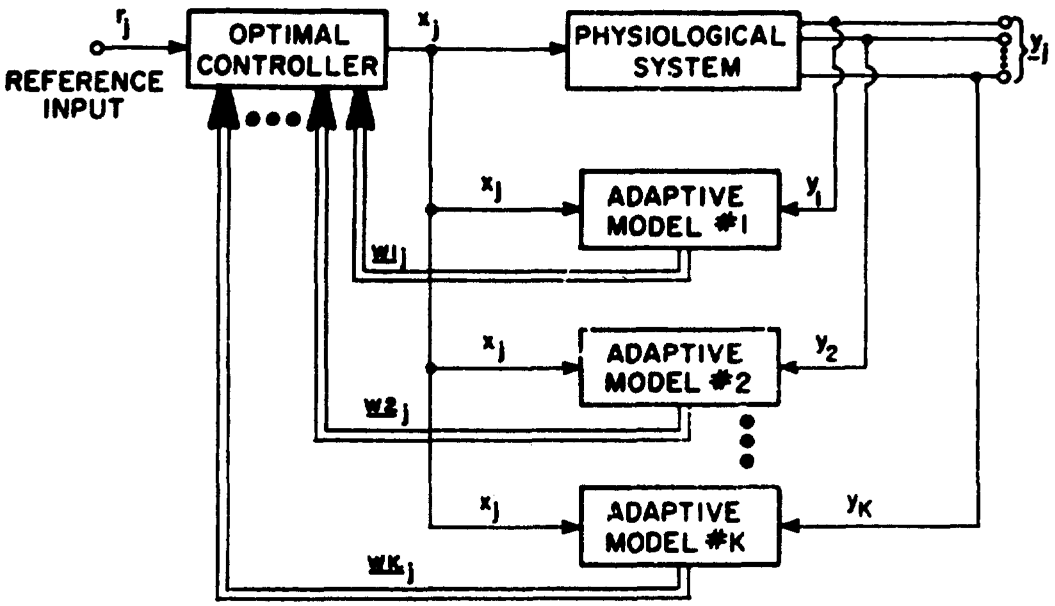


Fig. 16. Adaptive Multi-Model Controller.

in order to make use of more than one adaptive model. One approach would be to use the model that gives the "best" input-output relation for the control, and to use the remaining models as constraints only. Thus, the physician could not only specify what the blood pressure should be, but he could also specify what range of values other body processes could take on.

An alternate approach would be to extend the work of Installé [22] to decide when a model should be added or dropped from the control loop.

A controller that looks very promising for handling nonlinear systems is the adaptive differential-model controller (see Fig. 17). This controller was proposed by Strom [48] who conjectured that it should provide superior performance because it models the "local" behavior of the system. The weakness of this approach is that we must look at "derivatives" to form the model. In practice, a combination of the adaptive model controller and the adaptive differential-model controller may prove the most useful. The first would provide a "global"

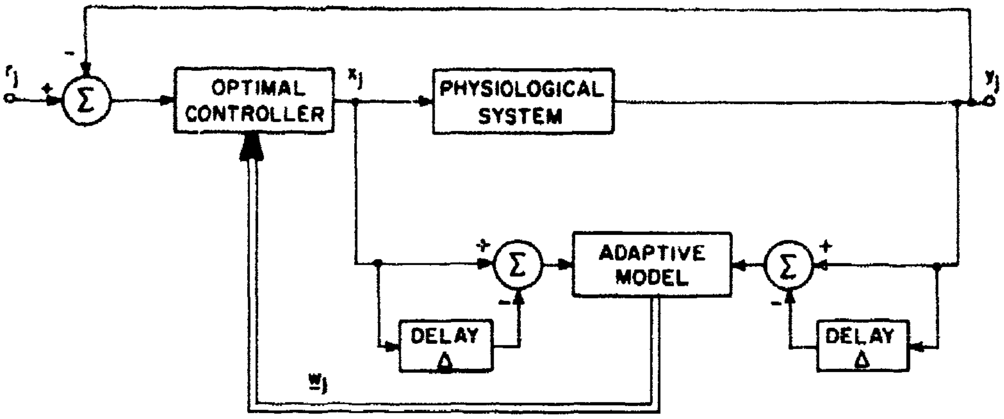


Fig. 17. Adaptive Differential-Model Controller.

model, while the second would give a "local" model. The model that is used to find the optimum input would then be a function of the magnitude of input.

The inverse-model controller (see Fig. 5) and/or the feedback model controller (see Fig. 6) might also be useful for controlling physiological systems. Comparatively little is known about these controllers, and hence more work in this area is certainly needed.

APPENDIX A

Norms of Vectors and Matrices

The notion of the distance between a vector and the origin is called a vector norm and satisfies the following properties [24,45]:

(1) $\|\underline{x}\| \geq 0$, $\|\underline{x}\| = 0$ iff $\underline{x} = 0$

(2) $\|c\underline{x}\| = |c|\ \|\underline{x}\|$, where c is any real number

(3) $\|\underline{x} + \underline{y}\| \leq \|\underline{x}\| + \|\underline{y}\|$.

We will use the Euclidean norm for vectors

$$\|\underline{x}\| = \left(\sum_{i=1}^{N} x_i^2 \right)^{1/2} .$$

Matrix norms have an additional property which is

(4) $\|\underline{\underline{A}}\,\underline{\underline{B}}\| \leq \|\underline{\underline{A}}\|\ \|\underline{\underline{B}}\|$.

A matrix norm is said to be compatible with a vector norm if for any vector $\underline{x}$

$$\|\underline{\underline{A}}\underline{x}\| \leq \|\underline{\underline{A}}\|\ \|\underline{x}\| . \qquad \text{(A.00)}$$

The Euclidean norm for matrices is

$$\|\underline{\underline{A}}\| = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 \right)^{1/2}$$

which is compatible with the vector Euclidean norm. We will make use of the fact that if $\underline{\underline{A}}$ is a symmetric matrix, then

$$\|\underline{\underline{A}}\| = \max_i |\lambda_i(\underline{\underline{A}})| . \qquad \text{(A.0)}$$

Proofs of Theorems 1, 2, 3 & 4.

Proof of Theorem 1.

Given

$$\underline{w}_{j+1} = \underline{w}_j + \frac{\alpha}{\|\underline{x}_j\|^2} \underline{x}_j \epsilon_j$$

where

$$\epsilon_j = \underline{x}_j^T (\underline{G} - \underline{w}_j)$$

subtract $\underline{G}$ from both sides

$$(\underline{w}_{j+1} - \underline{G}) = (\underline{w}_j - \underline{G}) + \frac{\alpha}{\|\underline{x}_j\|^2} \underline{x}_j \epsilon_j \quad ;$$

premultiplying by the transpose of the above we have

$$\|\underline{w}_{j+1} - \underline{G}\|^2 = (\underline{w}_j - \underline{G})^T (\underline{w}_j - \underline{G}) + \alpha \frac{(\underline{w}_j - \underline{G})^T \underline{x}_j \epsilon_j}{\|\underline{x}_j\|^2}$$

$$+ \alpha \frac{\epsilon_j \underline{x}_j^T (\underline{w}_j - \underline{G})}{\|\underline{x}_j\|^2} + \alpha^2 \frac{\epsilon_j \underline{x}_j^T \underline{x}_j \epsilon_j}{\|x_j\|^4}$$

Thus

$$\|\underline{w}_{j+1} - \underline{G}\|^2 = \|\underline{w}_j - \underline{G}\|^2 - \underbrace{\alpha(2 - \alpha)}_{> 0} \underbrace{\frac{\epsilon_j^2}{\|\underline{x}_j\|^2}}_{\geq 0}$$

therefore

$$\|\underline{w}_{j+1} - \underline{G}\| \leq \|\underline{w}_j - \underline{G}\|$$

where equality holds iff $\epsilon_j = 0$.■

This completes the proof of Theorem 1.

Proof of Theorem 2.

By Theorem 1 we know that for any N-dimensional input sequence $\{\underline{x}_j\}_o^\infty$ the distance between $\underline{w}_j$ and $\underline{G}$ is non-increasing. Geometrically this means that we can construct an N-dimensional sphere in E^N centered at the point $\underline{G}$ such that the point $\underline{w}_i$ will lie inside this sphere for all $i > j$. Thus one way to show that $\underline{w}_j$ converges to $\underline{G}$ is to show that there is convergence along all N of any set of orthogonal axes of E^N. To accomplish this we introduce the operator $L_j(\cdot)$ that operates on the input sequence $\{\underline{x}_j\}$ in groups of N to produce sequences of orthogonal inputs $\{\underline{\eta}_j\}$. Thus, for $j = 2N, 3N, \ldots$ let

$$L_j\left(\{\underline{x}_i\}_{j-N}^{j-1}\right) = \begin{cases} \underline{\eta}_{j-N} = \underline{x}_{j-N} & \\ \underline{\eta}_i = (\underline{x}_i^T \, \underline{e}_i)\, \underline{e}_i & i = j-N+1, j-N+2, \ldots j-1 \\ \text{where} & \\ \underline{e}_i = \dfrac{\underline{x}_i - \sum\limits_{i=j-N}^{j-1} \dfrac{\underline{\eta}_i^T \, \underline{x}_i}{\underline{\eta}_i^T \, \underline{\eta}_i} \underline{\eta}_i}{\left\| \underline{x}_i - \sum\limits_{i=j-N}^{j-1} \dfrac{\underline{\eta}_i^T \, \underline{x}_i}{\underline{\eta}_i^T \, \underline{\eta}_i} \underline{\eta}_i \right\|} & \end{cases}$$

$L_j(\cdot)$ performs the Gram-Schmidt orthogonalization on our linearly independent input sequence $\{\underline{x}_j\}$. Thus we can write $\underline{x}_j = \underline{\eta}_j + \underline{\tilde{x}}_j$ where $\underline{\eta}_j$ is the new information and is orthogonal to $\underline{\tilde{x}}_j$. Thus

$$(\underline{w}_{j+1} - \underline{G}) = \left[I - \alpha \frac{\underline{\eta}_j \underline{\eta}_j^T}{\|\underline{x}_j\|^2} - \alpha \frac{\underline{\tilde{x}}_j \underline{\tilde{x}}_j^T}{\|\underline{x}_j\|^2} \right] (\underline{w}_j - G) \ .$$

Premultiplying by the transpose of the above we have

$$\|\underline{w}_{j+1} - \underline{G}\|^2 = \left\| \left[I - \alpha \frac{\underline{\eta}_j \underline{\eta}_j^T}{\|\underline{x}_j\|^2} \right] (\underline{w}_j - \underline{G}) \right\|^2$$

$$- (\underline{w}_j - \underline{G})^T \left[\alpha \frac{\tilde{\underline{x}}_j \tilde{\underline{x}}_j^T}{\|\underline{x}_j\|^2} \right] \left[I - \alpha \frac{\underline{\eta}_j \underline{\eta}_j^T}{\|\underline{x}_j\|^2} \right] (\underline{w}_j - \underline{G})$$

$$- (\underline{w}_j - \underline{G})^T \left[I - \alpha \frac{\underline{\eta}_j \underline{\eta}_j^T}{\|\underline{x}_j\|^2} \right] \left[\alpha \frac{\tilde{\underline{x}}_j \tilde{\underline{x}}_j^T}{\|\underline{x}_j\|^2} \right] (\underline{w}_j - \underline{G})$$

$$+ \alpha^2 (\underline{w}_j - \underline{G})^T \left[\frac{\tilde{\underline{x}}_j \tilde{\underline{x}}_j^T}{\|\underline{x}_j\|^2} \right] \left[\frac{\tilde{\underline{x}}_j \tilde{\underline{x}}_j^T}{\|\underline{x}_j\|^2} \right] (\underline{w}_j - \underline{G})$$

$$= \left\| \left[I - \alpha \frac{\underline{\eta}_j \underline{\eta}_j^T}{\|\underline{x}_j\|^2} \right] (\underline{w}_j - \underline{G}) \right\|^2$$

$$- 2\alpha (\underline{w}_j - \underline{G})^T \left[\frac{\tilde{\underline{x}}_j \tilde{\underline{x}}_j^T}{\|\underline{x}_j\|^2} \right] (\underline{w}_j - \underline{G})$$

$$+ \alpha^2 \frac{\|\tilde{\underline{x}}_j\|^2}{\|\tilde{\underline{x}}_j\|^2} \underbrace{(\underline{w}_j - \underline{G})^T \left[\frac{\tilde{\underline{x}}_j \tilde{\underline{x}}_j^T}{\|\underline{x}_j\|^2} \right] (\underline{w}_j - \underline{G})}_{\text{non-negative definite}} \qquad \text{(A.1)}$$

Now looking at 2α and α^2 we see that

$$2\alpha \geq \alpha^2 \qquad \text{for} \qquad 0 < \alpha < 2$$

Thus the second term in (A.1) is always $\geq$ the last term. Therefore

$$\left\|\underline{w}_{j+1} - \underline{G}\right\|^2 \leq \left\| \left[I - \alpha \frac{\underline{\eta}_j \underline{\eta}_j^T}{\left\|\underline{x}_j\right\|^2} \right] (\underline{w}_j - \underline{G}) \right\|^2$$

where equality holds when $\tilde{\underline{x}}_j = 0$. The convergence due to the $\{\underline{\eta}_j\}$'s will thus form the desired upper bound. To find this upper bound ve need to look at the convergence due to $\{\underline{\eta}_i\}$ after N adaptions. Thus keeping tabs on only the convergence due to the new information at each adaption, we have

$$(\underline{w}_{j+1} - \underline{G}) = \left[I - \alpha \frac{\underline{\eta}_j \underline{\eta}_j^T}{\left\|\underline{x}_j\right\|^2} \right] (\underline{w}_j - \underline{G}) \qquad \text{(A.2)}$$

$$(\underline{w}_{j+2} - \underline{G}) = \left[I - \alpha \frac{\underline{\eta}_{j+1} \underline{\eta}_{j+1}^T}{\left\|\underline{x}_{j+1}\right\|^2} \right] (\underline{w}_{j+1} - \underline{G}) - \underline{o}_{j+1}$$

$$= \left[I - \alpha \frac{\underline{\eta}_{j+1} \underline{\eta}_{j+1}^T}{\left\|\underline{x}_{j+1}\right\|^2} \right] \left\{ \left[I - \alpha \frac{\underline{\eta}_j \underline{\eta}_j^T}{\left\|\underline{x}_j\right\|^2} \right] (\underline{w}_j - \underline{G}) \right\} - \underline{o}_{j+1}$$

where

$$\underline{o}_i \triangleq \text{[terms involving } \tilde{\underline{x}}_i \text{ as in (A.1)]}$$

$$(\underline{w}_{j+2} - \underline{G}) = \left[I - \alpha \left(\frac{\underline{x}_{j+1}\underline{x}_{j+1}^{T}}{\|\underline{x}_{j+1}\|^{2}} + \frac{\underline{x}_{j}\underline{x}_{j}^{T}}{\|\underline{x}_{j}\|^{2}} \right) \right] (\underline{w}_j - \underline{G}) - O_{j+1}$$

$$\vdots$$

$$(\underline{w}_{j+N} - \underline{G}) = \left[I - \alpha \sum_{i=j}^{j+N-1} \frac{\underline{x}_{i}\underline{x}_{i}^{T}}{\|\underline{x}_{i}\|^{2}} \right] (\underline{w}_j - \underline{G}) - \underline{O}_{j+N-1}$$

Thus for all $j=N,2N,3N, \ldots$ we have

$$(\underline{w}_{j} - \underline{G}) = \left[I - \alpha \sum_{i=j-N}^{j-1} \frac{\underline{x}_{i}\underline{x}_{i}^{T}}{\|\underline{x}_{i}\|^{2}} \right] (\underline{w}_{j-N} - \underline{G}) - \underline{O}_{j-1}$$

Premultiplying by the transpose of the above we have

$$\|\underline{w}_{j} - \underline{G}\|^{2} = \left\| \left[I - \alpha \sum_{i=j-N}^{j-1} \frac{\underline{x}_{i}\underline{x}_{i}^{T}}{\|\underline{x}_{i}\|^{2}} \right] (\underline{w}_{j-N} - \underline{G}) - \underline{O}_{j-1} \right\|^{2} \qquad \text{(A.3)}$$

Now given that $0 < \alpha < 2$, we have by iteration of the results of (A.1)

$$\|\underline{w}_{j} - \underline{G}\|^{2} \leq \left\| \left[I - \alpha \sum_{i=j-N}^{j-1} \frac{\underline{x}_{i}\underline{x}_{i}^{T}}{\|\underline{x}_{i}\|^{2}} \right] (\underline{w}_{j-N} - \underline{G}) \right\|^{2}$$

and by using (A.00), we have

$$\leq \left\| I - \alpha \sum_{i=j-N}^{j-1} \frac{\underline{x}_{i}\underline{x}_{i}^{T}}{\|\underline{x}_{i}\|^{2}} \right\|^{2} \|\underline{w}_{j-N} - \underline{G}\|^{2}$$

Let

$$\underline{\underline{A}}_j = \left[I - \alpha \sum_{i=j-N}^{j-1} \frac{\eta_i \eta_i^T}{\left\| \underline{x}_i \right\|^2} \right] \qquad \text{(A.4)}$$

Thus

$$\left\| \underline{w}_j - \underline{G} \right\|^2 \leq \left\| \underline{\underline{A}}_j \right\|^2 \left\| \underline{w}_{j-N} - \underline{G} \right\|^2 \qquad j = 2N, 3N, \ldots$$

Now to find out what $\left\| \underline{\underline{A}}_j \right\|^2$ is equal to, we solve for the eigenvalues of $\underline{\underline{A}}_j$. Assume that $\underline{c}_j$ is an eigenvector. Then

$$\left(I - \alpha \sum_{i=j-N}^{j-1} \frac{\eta_i \eta_i^T}{\left\| \underline{x}_i \right\|^2} \right) \underline{c}_j = \lambda_j \underline{c}_j$$

Thus we find that

$$\underline{c}_j = \frac{\eta_j}{\left\| \eta_j \right\|}$$

and

$$\lambda_j = 1 - \alpha \frac{\left\| \eta_j \right\|^2}{\left\| \underline{x}_j \right\|^2}$$

and since

$$\delta < \frac{\left\| \eta_j \right\|^2}{\left\| \underline{x}_j \right\|^2} \leq 1$$

and

$$0 < \alpha < 2$$

we have that

$$-1 < \lambda_j < 1$$

let

$$b_j = \max_{j-N<i<j-1} \lambda_j$$

Then

$$-1 < b_j < 1 \qquad \text{for all} \qquad j = 2N, 3N, \ldots$$

Now we can write

$$\|\underline{w}_N - \underline{G}\|^2 \leq b_o^2 \|\underline{w}_o - \underline{G}\|^2$$

$$\|\underline{w}_{2N} - \underline{G}\|^2 \leq b_N^2 \|\underline{w}_N - \underline{G}\|^2 \leq b_N^2 b_o^2 \|\underline{w}_o - \underline{G}\|^2$$

$$\vdots$$

$$\|\underline{w}_j - \underline{G}\|^2 \leq \prod_{i=o}^{j} b_i^2 \|\underline{w}_o - \underline{G}\|^2 \qquad j = 2N, 3N, \ldots \qquad \text{(A.5)}$$

Let

$$b_{max}^2 = \limsup b_i^2$$

then

$$b_{max}^2 < (1 - \alpha\delta)^2$$

where $\delta > 0$ by definition of an N-dimensional input sequence; and since

$$\lim_{n \to \infty} (1 - \alpha\delta)^{2n} = 0$$

we have that

$$\lim_{j \to \infty} \prod_{i=o}^{j} b_i^2 \|\underline{w}_o - \underline{G}\|^2 = 0$$

Therefore

$$\lim_{j \to \infty} \underline{w}_j = \underline{G} \quad . \blacksquare$$

<u>This completes the proof of Theorem 2.</u>

Proof of Theorem 3.

Given

$$y_j = \underline{x}_j^T \underline{G} + n_j$$

we have that

$$\underline{w}_{j+1} = \underline{w}_j + \frac{\alpha}{\|\underline{x}_j\|^2}\, \underline{x}_j \left(\underline{x}_j^T \underline{G} + n_j - \underline{x}_j^T \underline{w}_j\right)$$

Subtracting $\underline{G}$ from both sides we have

$$(\underline{w}_{j+1} - \underline{G}) = \left(I - \alpha \frac{\underline{x}_j \underline{x}_j^T}{\|\underline{x}_j\|^2}\right) (\underline{W}_j - \underline{G}) - \alpha \frac{\underline{x}_j n_j}{\|\underline{x}_j\|^2} \tag{A.6}$$

Taking the expected value with respect to the noise source we have (recall that n_j is zero mean)

$$(\overline{\underline{w}}_{j+1} - \underline{G}) = \left(I - \alpha \frac{\underline{x}_j \underline{x}_j^T}{\|\underline{x}_j\|^2}\right) (\overline{\underline{w}}_j - \underline{G})$$

Now, substituting $\overline{\underline{w}}_{j+1}$ for $\underline{w}_{j+1}$ and $\overline{\underline{w}}_j$ for $\underline{w}_j$ in (A.2) we have

$$\lim_{j \to \infty} \overline{\underline{w}}_j = \underline{G} \quad . \blacksquare$$

This completes the first part of the proof.

Let

$$\underline{\underline{D}}_j = I - \alpha \frac{\underline{x}_j \underline{x}_j^T}{\|\underline{x}_j\|^2} \qquad \text{and} \qquad \gamma_j = \alpha \frac{\underline{x}_j n_j}{\|\underline{x}_j\|^2}$$

Then (A.6) is

$$(\underline{w}_{j+1} - \underline{G}) = \underline{\underline{D}}_j (\underline{w}_j - \underline{G}) - \underline{\gamma}_j \tag{A.7}$$

and we have for an N step

$$(\underline{w}_{j+2} - \underline{G}) = \underline{\underline{D}}_{j+1} (\underline{w}_{j+1} - \underline{G}) - \underline{\gamma}_{j+1}$$

$$= \underline{\underline{D}}_{j+1} (\underline{\underline{D}}_j (\underline{w}_j - \underline{G}) - \underline{\gamma}_j) - \underline{\gamma}_{j+1}$$

$$= \underline{\underline{D}}_{j+1} \underline{\underline{D}}_j (\underline{w}_j - \underline{G}) - \underline{\underline{D}}_{j+1} \underline{\gamma}_j - \underline{\gamma}_{j+1}$$

$$(\underline{w}_{j+3} - \underline{G}) = \underline{\underline{D}}_{j+2} (\underline{w}_{j+2} - \underline{G}) - \underline{\gamma}_{j+2}$$

$$= \underline{\underline{D}}_{j+2} (\underline{\underline{D}}_{j+1} \underline{\underline{D}}_j (\underline{w}_j - \underline{G}) - \underline{\underline{D}}_{j+1} \underline{\gamma}_j - \underline{\gamma}_{j+1}) - \underline{\gamma}_{j+2}$$

$$= \underline{\underline{D}}_{j+2} \underline{\underline{D}}_{j+1} \underline{\underline{D}}_j (\underline{w}_j - \underline{G}) - \underline{\underline{D}}_{j+2} \underline{\underline{D}}_{j+1} \underline{\gamma}_j - \underline{\underline{D}}_{j+2} \underline{\gamma}_{j+1} - \underline{\gamma}_{j+2}$$

$$\vdots$$

$$(\underline{w}_{j+N} - \underline{G}) = \prod_{k=j}^{j+N-1} \underline{\underline{D}}_k (\underline{w}_j - \underline{G}) - \sum_{i=1}^{N-1} \prod_{k=N-i}^{N-1} \underline{\underline{D}}_{j+k} \underline{\gamma}_{j+N-1-i} - \underline{\gamma}_{j+N-1} .$$

Thus for all $j = N, 2N, 3N, \ldots$, we have

$$(\underline{w}_j - \underline{G}) = \prod_{k=j-N}^{j-1} \underline{\underline{D}}_k (\underline{w}_{j-N} - \underline{G}) - \sum_{i=1}^{N-1} \prod_{k=N-i}^{N-1} \underline{\underline{D}}_{j-N+k} \underline{\gamma}_{j-1-i} - \underline{\gamma}_{j-1} \tag{A.8}$$

Premultiplying by the transpose of the above

$$\left\|\underline{w}_j - \underline{G}\right\|^2 = \left\| \prod_{k=j-N}^{j-1} \underline{\underline{D}}_k (\underline{w}_{j-N} - \underline{G}) \right\|^2$$

$$+ \left\|\underline{\gamma}_{j-1}\right\|^2 + \left\|\underline{\underline{D}}_{j-1}\underline{\gamma}_{j-2}\right\|^2 + \cdots + \left\| \prod_{k=j-N+2}^{j-1} \underline{\underline{D}}_k \underline{\gamma}_{j-N} \right\|^2$$

$$+ \text{cross terms involving } \underline{\gamma}_i\text{'s}$$

Taking the expected value with respect to the noise source, the cross terms involving $\underline{\gamma}$ drop out because the noise is zero mean, thus

$$\overline{\left\|\underline{w}_j - \underline{G}\right\|^2} = \overline{\left\| \prod_{k=j-N}^{j-1} \underline{\underline{D}}_k (\underline{w}_{j-N} - \underline{G}) \right\|^2}$$

$$+ \overline{\left\|\underline{\gamma}_{j-1}\right\|^2} + \overline{\left\|\underline{\underline{D}}_{j-1}\underline{\gamma}_{j-2}\right\|^2} + \cdots + \overline{\left\| \prod_{k=j-N+2}^{j-1} \underline{\underline{D}}_k \underline{\gamma}_{j-N} \right\|^2}$$

$$\leq \overline{\left\| \prod_{k=j-N}^{j-1} \underline{\underline{D}}_k (\underline{w}_{j-N+1} - \underline{G}) \right\|^2}$$

$$+ \overline{\left\|\underline{\gamma}_{j-1}\right\|^2} + \left\|\underline{\underline{D}}_{j-1}\right\|^2 \overline{\left\|\underline{\gamma}_{j-2}\right\|^2} + \cdots + \left\| \prod_{k=j-N+2}^{j-1} \underline{\underline{D}}_k \right\|^2 \overline{\left\|\underline{\gamma}_{j-N}\right\|^2} \qquad (A.9)$$

Now making use of (A.0), we have

$$\left\|\underline{\underline{D}}_{j-1}\right\|^2, \left\|\underline{\underline{D}}_{j-1}\underline{\underline{D}}_{j-2}\right\|^2, \ldots, \left\| \prod_{k=j-N+2}^{j-1} \underline{\underline{D}}_k \right\|^2 = \lambda_{max} = 1 \qquad (A.10)$$

Therefore

$$\overline{\|\underline{w}_j - \underline{G}\|^2} \le \overline{\left\| \prod_{k=j-N}^{j-1} \underline{\underline{D}}_k (\underline{w}_{j-N} - \underline{G}) \right\|^2} + \sum_{k=1}^{N} \overline{\|\underline{\gamma}_{j-k}\|^2}$$

Noticing that

$$\overline{\|\underline{\gamma}_j\|^2} = \overline{\left\| \frac{\alpha \underline{x}_j n_j}{\|\underline{x}_j\|^2} \right\|^2} = \alpha^2 \overline{n_j^2} \, \frac{\underline{x}_j^T \underline{x}_j}{\|\underline{x}_j\|^4} = \frac{\alpha^2 \sigma^2}{\|\underline{x}_j\|^2}$$

we can write

$$\overline{\|\underline{w}_j - \underline{G}\|^2} \le \overline{\left\| \prod_{k=j-N}^{j-1} \underline{\underline{D}}_k (\underline{w}_{j-N} - \underline{G}) \right\|^2} + \alpha^2 \sigma^2 \sum_{k=1}^{N} \frac{1}{\|\underline{x}_{j-k}\|^2}$$

Now we have

$$\overline{\|\underline{w}_j - \underline{G}\|^2} \le \overline{\left\| \prod_{k=j-N}^{j-1} \underline{\underline{D}}_k (\underline{w}_{j-N} - \underline{G}) \right\|^2} + \frac{N\alpha^2 \sigma^2}{\|\underline{x}_{\min}\|^2}$$

Now referring to (A.3) we see that

$$\overline{\left\| \prod_{k=j-N}^{j-1} \underline{\underline{D}}_k (\underline{w}_{j-N} - \underline{G}) \right\|^2}$$

can be written as

$$\overline{\left\| \left[I - \alpha \sum_{i=j-N}^{j-1} \frac{\underline{n}_i \underline{n}_i^T}{\|\underline{x}_i\|^2} \right] (\underline{w}_{j-N} - \underline{G}) - \underline{0}_{j-1} \right\|^2}$$

Thus, given that $0 < \alpha < 2$, we have

$$\overline{\|\underline{w}_j - \underline{G}\|^2} \le \|\underline{\underline{A}}_j\|^2 \overline{\|\underline{w}_{j-N} - \underline{G}\|^2} + \frac{N\alpha^2 \sigma^2}{\|\underline{x}_{\min}\|^2}$$

where $\underline{\underline{A}}_j$ is as defined in (A.4). Now to get an equation similar to (A.5) we do the following

$$\overline{\|\underline{w}_N - \underline{G}\|^2} \leq b_o^2 \overline{\|\underline{w}_o - \underline{G}\|^2} + \omega \qquad \text{(A.11)}$$

where

$$\omega = \frac{N\alpha^2 \sigma^2}{\|\underline{x}_{min}\|^2}$$

$$\overline{\|\underline{w}_{2N} - \underline{G}\|^2} \leq b_N^2 \overline{\|\underline{w}_N - \underline{G}\|^2} + \omega$$

$$\leq b_N^2 b_o^2 \overline{\|\underline{w}_o - \underline{G}\|^2} + (b_N^2 + 1)\omega$$

$$\overline{\|\underline{w}_{3N} - \underline{G}\|^2} \leq b_{2N}^2 \overline{\|\underline{w}_{2N} - \underline{G}\|^2} + \omega$$

$$\leq b_{2N}^2 b_N^2 b_o \overline{\|\underline{w}_o - \underline{G}\|^2} + (b_{2N}^2 b_N^2 + b_{2N}^2 + 1)\omega$$

$$\vdots$$

$$\overline{\|\underline{w}_j - \underline{G}\|^2} \leq \prod_{i=o}^{j} b_i^2 \overline{\|\underline{w}_o - \underline{G}\|^2} + \left[\sum_{i=1}^{j} \prod_{k=j+1-i}^{j} b_k^2 + 1 \right] \omega \qquad j = 2N, 3N, \ldots$$

Now substituting b_{max} for b_i we get

$$\overline{\|\underline{w}_j - \underline{G}\|^2} \leq b_{max}^{2j} \overline{\|\underline{w}_o - \underline{G}\|^2} + \omega \sum_{i=o}^{j} b_{max}^{2i}$$

Now since

$$\lim_{j \to \infty} b_{\max}^{2j} = 0$$

and

$$\lim_{j \to \infty} \sum_{i=0}^{j} b_{\max}^{2i} = \frac{1}{1 - b_{\max}^{2}}$$

we have substituting for ω and taking the limit

$$\lim_{j \to \infty} \overline{\|\underline{w}_j - \underline{G}\|^2} \le \frac{\alpha^2 \sigma^2 N}{(1 - b_{\max}^2)\|\underline{x}_{\min}\|^2}$$

$$\le \frac{\alpha \sigma^2 N}{\delta(2-\alpha\delta)\|\underline{x}_{\min}\|^2}$$

and using the Cauchy-Schwartz inequality we have

$$\underline{x}_j^T(\underline{w}_j - \underline{G}) \le \|\underline{x}_j\|\|\underline{w}_j - \underline{G}\| \le \|\underline{x}_{\max}\|\|\underline{w}_j - \underline{G}\|$$

and thus

$$\lim_{j \to \infty} \overline{\left[\underline{x}_j^T(\underline{w}_j - \underline{G})\right]^2} \le \frac{\alpha\ \sigma^2 N\|\underline{x}_{\max}\|^2}{\delta(2 - \alpha\delta)\|\underline{x}_{\min}\|^2} \quad . \blacksquare$$

<u>This completes the proof of Theorem 3.</u>

Proof of Theorem 4.

Given

$$\underline{w}'_{j+1} = \underline{w}'_j + \frac{\alpha \underline{x}'_j}{\|\underline{x}'_j\|^2}\left(\underline{x}_j^T \underline{G} + n_j - \underline{x}'^T_j \underline{w}'_j\right)$$

we have

$$\underline{w}'_{j+1} = \underline{w}'_j + \frac{\alpha \underline{x}'_j}{\|\underline{x}'_j\|^2}\,\underline{x}'^T_j \left[\begin{array}{c} n_j - w_{oj} \\ \hline \underline{G} - \underline{w}_j \end{array}\right] \qquad (A.12)$$

where w_{oj} is weight zero at time j; or in partitioned form

$$\left[\begin{array}{c} w_{oj+1} \\ \hline \underline{w}_{j+1} \end{array}\right] = \left[\begin{array}{c} w_{oj} \\ \hline \underline{w}_j \end{array}\right] + \left[\frac{\alpha}{\|\underline{x}_j\|^2 + 1}\right]\left[\begin{array}{c|c} 1 & \underline{x}_j^T \\ \hline \underline{x}_j & \underline{x}_j\underline{x}_j^T \end{array}\right]\left[\begin{array}{c} n_j - w_{oj} \\ \hline \underline{G} - \underline{w}_j \end{array}\right]$$

Taking the expected value with respect to the noise source we have

$$\bar{\underline{w}}'_{j+1} = \bar{\underline{w}}'_j + \alpha\,\frac{\underline{x}'_j\underline{x}'^T_j}{\|\underline{x}'_j\|^2}\left[\begin{array}{c} \beta - \bar{w}_{oj} \\ \hline \underline{G} - \bar{\underline{w}}_j \end{array}\right]$$

Now, subtracting $\underline{G}'$ from both sides where

$$\underline{G}' = \left[\begin{array}{c} \beta \\ \hline \underline{G} \end{array}\right]$$

we have

$$(\bar{\underline{w}}'_{j+1} - \underline{G}') = \left[I - \alpha\,\frac{\underline{x}'_j\underline{x}'^T_j}{\|\underline{x}'_j\|^2}\right](\bar{\underline{w}}_j - \underline{G}')$$

Now, making the appropriate substitutions for w, G and x in (A.2) we have

$$\lim_{j \to \infty}\|\bar{\underline{w}}'_j - \underline{G}'\|^2 = 0$$

or equivalently

$$\lim_{j \to \infty} \underline{w}_{oj} = \beta$$

and

$$\lim_{j \to \infty} \underline{\bar{w}}_{j} = \underline{G} \quad . \blacksquare$$

This completes the first part of the proof.

Letting

$$\underline{\underline{D}}'_j = \left[I - \alpha \frac{\underline{x}'_j \underline{x}'^T_j}{\|\underline{x}'_j\|^2} \right]$$

$$\underline{G}'_j = \begin{bmatrix} n_j \\ - - - \\ \underline{G} \end{bmatrix}$$

$$\underline{\gamma}'_j = \begin{bmatrix} n_j - n_{j+1} \\ - - - - - \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

we can write (A.12) as follows

$$\underline{w}'_{j+1} - \begin{bmatrix} n_{j+1} \\ - - - - \\ \underline{G} \end{bmatrix} = \underline{w}'_{j+1} - \begin{bmatrix} n_j \\ - - - \\ \underline{G} \end{bmatrix} - \alpha \frac{\underline{x}'_j \underline{x}'^T_j}{\|\underline{x}'_j\|^2} (\underline{w}'_j - \underline{G}'_j) + \underline{\gamma}'_j$$

or

$$(\underline{w}'_{j+1} - \underline{G}'_{j+1}) = \underline{\underline{D}}'_j (\underline{w}'_j - \underline{G}') + \underline{\gamma}'_j$$

Now substituting into (A.7), we have for an N step

$$(\underline{w}'_j - \underline{G}'_j) = \prod_{k=j-N}^{j-1} \underline{\underline{D}}'_k (\underline{w}'_{j-N} - \underline{G}'_{j-N}) + \sum_{i=1}^{N-1} \prod_{k=N-i}^{N-1} \underline{\underline{D}}'_{j-N+k} \underline{\gamma}'_{j-1-i} + \underline{\gamma}'_{j-1}$$

for all $j = N, 2N, 3N, \ldots$

Premultiplying by the transpose of the above

$$\|\underline{w}'_j - \underline{G}'_j\|^2 = \left\| \prod_{k=j-N}^{j-1} \underline{\underline{D}}'_k (\underline{w}'_{j-N} - \underline{G}'_{j-N}) \right\|^2$$

$$+ \|\underline{\gamma}'_{j-1}\|^2 + \|\underline{\underline{D}}'_{j-1}\underline{\gamma}'_{j-2}\|^2 + \dots + \left\| \prod_{k=j-N+2}^{j-1} \underline{\underline{D}}'_k \underline{\gamma}'_{j-N} \right\|^2$$

$$+ 2(\underline{w}'_{j-N} - \underline{G}'_{j-N})^T \left[\prod_{k=j-N}^{j-1} \underline{\underline{D}}'_k \right]^T \left(\sum_{i=1}^{N-1} \prod_{k=N-i}^{N-1} \underline{\underline{D}}'_{j-N+k} \underline{\gamma}'_{j-1-i} + \underline{\gamma}'_{j-1} \right)$$

$$+ (N^2 - N \text{ cross terms involving } \underline{\gamma}'_i, \underline{\gamma}'_k, i \neq k) \qquad (A.13)$$

Consider individually the expected value with respect to the noise source of the four terms on the right.

1) The first term of (A.13) is (referring to (A.3))

$$\overline{\left\| \prod_{k=j-N}^{j-1} \underline{\underline{D}}'_k (\underline{w}'_{j-N} - \underline{G}'_{j-N}) \right\|^2} \leq \|\underline{\underline{A}}'_j\|^2 \overline{\|\underline{w}'_{j-N} - \underline{G}'_{j-N}\|^2}$$

where

$$\underline{\underline{A}}'_j = \left[I - \alpha \sum_{i=j-N}^{j-1} \frac{\underline{n}'_i \underline{n}'^T_i}{\|\underline{x}'_i\|^2} \right]$$

2) Consider the second term of (A.13):

$$\overline{\|\underline{\gamma}'_{j-1}\|^2} + \overline{\|\underline{\underline{D}}'_{j-1}\underline{\gamma}'_{j-2}\|^2} + \dots + \overline{\left\| \prod_{k=j-N+2}^{j-1} \underline{\underline{D}}'_k \underline{\gamma}'_{j-N} \right\|^2}$$

referring to (A.9) and (A.10), we have

$$\text{second term} \leq \sum_{k=1}^{N} \overline{\left\| \underline{\gamma}'_{j-k} \right\|^2}$$

but recall that

$$\underline{\gamma}_i'^{T} \underline{\gamma}_i' = \overline{(n_i - n_{i+1})^2}$$

$$= \overline{n_i^2} - 2\beta^2 + \overline{n_{i+1}^2}$$

$$= 2\sigma^2$$

Therefore

$$\text{second term} \leq 2N\sigma^2$$

3) Consider the third term of (A.13):

$$\overline{2(\underline{w}'_{j-N} - \underline{G}'_{j-N})^T \left[\prod_{k=j-N}^{j-1} \underline{\underline{D}}'_k \right]^T \left(\sum_{i=1}^{N-1} \prod_{k=N-i}^{N-1} \underline{\underline{D}}'_{j-N+k} \underline{\gamma}'_{j-1-i} + \underline{\gamma}'_{j-1} \right)}$$

Note that only the $\underline{\gamma}'_{j-N}$ term is correlated with $\underline{G}'_{j-N}$; thus all the other terms are equal to zero because $\overline{\underline{\gamma}'_i} = [0\ 0\ .\ .\ .\ 0]^T$. Thus

$$\text{third term} = \overline{2(\underline{w}'_{j-N} - \underline{G}'_{j-N})^T \left[\prod_{k=j-N}^{j-1} \underline{\underline{D}}'_k \right]^T \left(\prod_{k=1}^{N-1} \underline{\underline{D}}'_{j-N+k} \underline{\gamma}'_{j-N} \right)}$$

or

$$= \overline{2(\underline{w}'_{j-N} - \underline{G}'_{j-N})^T \underline{\underline{DD}}_j \underline{\gamma}'_{j-N}}$$

where

$$\underline{\underline{DD}}_j = \left[\prod_{k=j-N}^{j-1} \underline{\underline{D}}'_k \right]^T \left[\prod_{k=1}^{N-1} \underline{\underline{D}}'_{j-N+k} \right]$$

In expanded form we have

$$\text{third term} = \overline{2(n_{j-N} - n_{j-N+1})(\underline{w}'_{j-N} - \underline{G}'_{j-N})^T}\,\underline{\underline{DD}}_j\underline{Z}$$

where

$$\underline{Z} = [100\ .\ .\ .\ 0]^T$$

The $\underline{w}'_{j-N}$ vector drops out because it is uncorrelated with n_{j-N} and n_{j-N+1} and is therefore multiplied by zero. Similarly, only the first term of $\underline{G}'_{j-N}$ in non-zero. Thus

$$\text{third term} = -2\overline{(n_{j-N} - n_{j-N+1})(n_{j-N})}\ \underline{Z}^T\underline{\underline{DD}}_j\underline{Z}$$

$$= -2(\overline{n^2_{j-N}} - \beta^2)\ \underline{Z}^T\underline{\underline{DD}}_j\underline{Z}$$

$$= -2\sigma^2\underline{Z}^T\underline{\underline{DD}}_j\underline{Z}$$

Now, making use of (A.00), we have

$$\text{third term} \leq 2\sigma^2|\underline{Z}^T\underline{\underline{DD}}_j\underline{Z}|$$

$$\leq 2\sigma^2\|\underline{Z}\|\ \|\underline{\underline{DD}}_j\underline{Z}\|$$

$$\leq 2\sigma^2\|\underline{Z}\|\ \|\underline{\underline{DD}}_j\|\ \|\underline{Z}\| \qquad . \qquad \text{(A.14)}$$

but

$$\|\underline{Z}\| = 1$$

and

$$\|\underline{\underline{DD}}_j\| = 1$$

therefore

$$\text{third term} \leq 2\sigma^2$$

4) Consider the final term of (A.13):

$$\overline{(N^2 - N \text{ cross terms involving } \underline{\gamma}_i',\ \underline{\gamma}_k',\ i \neq k)}$$

Notice that these cross terms can be written in the form

$$\overline{(n_i - n_{i+1})(n_k - n_{k+1})}\ \underline{Z}^T \underline{\underline{M}}\,\underline{Z} \quad .$$

where $\underline{\underline{M}}$ is a matrix corresponding to the product of the $\underline{\underline{D}}_j'$'s. Only $2(N-1)$ of these terms are non-zero. These are the terms where $|i - k| = 1$. For each of these terms we have (see (A.14) and following)

$$\begin{aligned}\overline{(n_i - n_{i+1})(n_k - n_{k+1})}\ \underline{Z}^T \underline{\underline{M}}\,\underline{Z} &= -\sigma^2 \underline{Z}^T \underline{\underline{M}}\,\underline{Z} \\ &\leq \sigma^2 \|\underline{Z}\|\ \|\underline{\underline{M}}\|\ \|\underline{Z}\| \\ &\leq \sigma^2\end{aligned}$$

Therefore

$$\text{fourth term} \leq 2(N-1)\sigma^2$$

We have now proven that

$$\overline{\|\underline{w}_j' - \underline{G}_j'\|^2} \leq \|\underline{\underline{A}}_j'\|^2\, \overline{\|\underline{w}_{j-N}' - \underline{G}_{j-N}'\|^2} + 4\sigma^2 N$$

for all $j = N, 2N, 3N, \ldots$

Now substituting into (A.11), we have the desired result

$$\lim_{j \to \infty} \overline{\left[\underline{x}_j'^T (\underline{w}_j' - \underline{G}_j')\right]^2} \leq \frac{4\sigma^2 N \|\underline{x}_{max}'\|^2}{\alpha\delta(2 - \alpha\delta)} \quad . \blacksquare$$

This completes the proof of Theorem 4.

APPENDIX B

A General Description of the Algorithm Used in Phase 4.

In the last decade many methods have been developed for minimizing the quadratic objective function (3.8) subject to linear inequality constraints (3.10) [40-44]. One of the most promising is given by Goldfarb. His algorithm extends Davidon's variable metric technique for constrained minimization and is based on ideas found in Rosen [41]. When the Hessian is I, as it is in our case, Goldfarb's algorithm is equivalent to Rosen's Gradient Projection Algorithm.

The algorithm presented here is specifically designed for the case of a quadratic objective function (3.8), linear inequality constraints (3.10), the Hessian matrix of second partial derivatives being equal to the identity matrix (3.11) and the gradient being equal to z.

The algorithm described in Phase 4, which incorporates the ideas of Rosen and Gill and Murray [47], was worked out by Kaufman [46].

We want to minimize

$$f(\underline{z}^k) = \tfrac{1}{2}(\underline{z}^{k^T}\underline{z}^k)$$

subject to the constraints

$$\underline{n}_m^T\underline{z}^k \geq \ell_m \qquad m = 1,2, \ldots 4LN$$

where

$$\underline{n}_m^T\underline{n}_m = 1 \qquad m = 1,2, \ldots 4LN$$

A necessary and sufficient condition for $f(\underline{z}^*)$ to be the global minimum is that there exists an $\underline{\alpha}$ such that

$$\underline{g}(\underline{z}^*) = \underline{\underline{N}}_q\underline{\alpha}$$

where

$$\underline{\underline{N}}_q = \{\underline{n}_1, \underline{n}_2, \dots \underline{n}_q\}$$

The columns of $\underline{\underline{N}}_q$ are the q unit normals to the hyperplanes whose intersection is an affine subspace in which $\underline{z}^*$ lies. The iteration equation is thus

$$\underline{z}^{k+1} = \underline{z}^k - \underline{\underline{\hat{P}}}_q \underline{z}^k \qquad (B.1)$$

where

$$\underline{\underline{\hat{P}}}_q = I - \underline{\underline{N}}_q (\underline{\underline{N}}_q^T \underline{\underline{N}}_q)^{-1} \underline{\underline{N}}_q^T \qquad (B.2)$$

is Rosen's projection operator that projects E^m into the constraint manifold. Note that in the unconstrained case q equals zero and $\underline{\underline{\hat{P}}}_q = I$ and thus (B.1) reduces to the one-step Newton method, $\underline{z}^{k+1} = \underline{0}$. Substituting for $\underline{\underline{\hat{P}}}_q$ in (B.1) we have

$$\underline{z}^{k+1} = \underline{\underline{N}}_q (\underline{\underline{N}}_q^T \underline{\underline{N}}_q)^{-1} \underline{\underline{N}}_q^T \underline{z}^k$$

or equivalently

$$\underline{z}^{k+1} = \underline{\underline{N}}_q \underline{\alpha} \qquad (B.3)$$

where

$$\underline{\alpha} = (\underline{\underline{N}}_q^T \underline{\underline{N}}_q)^{-1} \underline{\ell}_q \qquad (B.4)$$

The elements of $\underline{\ell}_q$ correspond to the q active constraints of $\underline{\ell}$. The computation of $\underline{\alpha}$ can be quickly performed as two back solve operations if the following construction is employed. Assume that $\underline{\underline{N}}_q^T$ is $q \times t$, then $\underline{\underline{N}}_q$ can be written as $\underline{\underline{N}}_q = \underline{\underline{QP}}_q$ where $\underline{\underline{Q}}^T \underline{\underline{Q}} = \underline{\underline{QQ}}^T = \underline{\underline{I}}$ and $\underline{\underline{P}}_q$ is a $q \times q$ upper triangular matrix [45]. Thus, (B.4) can be written as

$$\underline{\underline{N}}_q^T \underline{\underline{N}}_q \underline{a} = \underline{\ell}_q$$

$$\underline{\underline{P}}_q^T \underline{\underline{Q}}^T \underline{\underline{Q}}\, \underline{\underline{P}}_q \underline{a} = \underline{\ell}_q$$

$$\underline{\underline{P}}_q^T \underline{\underline{P}}_q \underline{a} = \underline{\ell}_q$$

Since $\underline{\underline{P}}_q$ is an upper triangular matrix, _ can be quickly calculated by using two back solve operations,

$$\underline{\underline{P}}_q^T \underline{\beta} = \underline{\ell}_q$$

and

$$\underline{\underline{P}}_q \underline{a} = \underline{\beta} \quad .$$

The addition of constraints to $\underline{\underline{N}}_q$ is done as follows:

Let $\underline{\underline{N}}_{q+1} = \left[\underline{\underline{N}}_q \;\middle|\; \underline{n}_q\right]$. If $\underline{\underline{N}}_q = \underline{\underline{Q}}\, \underline{\underline{P}}_q$, then

$$\underline{\underline{N}}_{q+1}^T \underline{\underline{N}}_{q+1} = \left[\begin{array}{c|c} \underline{\underline{N}}_q^T \underline{\underline{N}}_q & \underline{\underline{N}}_q^T \underline{n}_q \\ \hline \underline{n}_q^T \underline{\underline{N}}_q & \underline{n}_q^T \underline{n}_q \end{array}\right]$$

$$= \left[\begin{array}{c|c} \underline{\underline{P}}_q^T \underline{\underline{P}}_q & \underline{\underline{N}}_q^T \underline{n}_q \\ \hline \underline{n}_q^T \underline{\underline{N}}_q & \underline{n}_q^T \underline{n}_q \end{array}\right]$$

If we put $\underline{\underline{P}}_{q+1}$ in upper triangular form, then

$$\underline{\underline{P}}_{q+1} = \left[\begin{array}{c|c} \underline{\underline{P}}_q & \underline{\gamma} \\ \hline 0 & p \end{array}\right]$$

and

$$\underline{\underline{N}}^T_{q+1}\underline{\underline{N}}_{q+1} = \underline{\underline{P}}^T_{q+1}\underline{\underline{P}}_{q+1} = \left[\begin{array}{c|c} \underline{\underline{P}}^T_q\underline{\underline{P}}_q & \underline{\underline{P}}^T_q\,\underline{\gamma} \\ \hline \underline{\gamma}^T\underline{\underline{P}}_q & \underline{\gamma}^T\underline{\gamma} + p^2 \end{array}\right]$$

Thus to calculate $\underline{\underline{P}}_{q+1}$ we need to solve for $\underline{\gamma}$ and p where

$$\underline{\underline{P}}^T_q\underline{\gamma} = \underline{\underline{N}}^T_q\underline{n}_q$$

and

$$p = (\underline{n}^T_q\underline{n}_q - \underline{\gamma}^T\underline{\gamma})^{\frac{1}{2}}$$

$$= (1 - \underline{\gamma}^T\underline{\gamma})^{\frac{1}{2}}$$

When a constraint is dropped the corresponding column must be deleted from $\underline{\underline{P}}_q$. If the k^{th} column is deleted, we have $q - k$ elements below the diagonal which must be set to zero. This is easily done using Given's rotation matrices [45].

The author suggests that interested readers see [40] Section 8 and [46] for a discussion of rates of convergence and operation counts.

APPENDIX C

A PROGRAMMED VERSION OF PHASE FOUR

```
REAL PROCEDURE PHASE4(N,LN,W,XØ,R,A,B,C,D); VALUE N,LN;
 INTEGER N,LN; REAL ARRAY W,XØ,R,A,B,C,D;
BEGIN
 REAL T1,T2,T3,T4,LSTAR,ASTAR,NORM2S,NORM2Z;
 INTEGER Q,J,K,N1,J1,J2,J3,I1,I2,I3,INDEX,START;
                                    & PDIM♡[(LN(LN+1))/2]
 EQUATE MAXLN♡40,LN4♡160,XV♡1,Ø'-2,PDIM♡820,EPSILON♡Ø.25'-5;
 REAL ARRAY HR,WIN,Z,S,NORM,ALPHA,BETA[1:MAXLN],P[1:PDIM],
       L[1:LN4];
 INTEGER ARRAY NQ[1:MAXLN];
 INTEGER PROCEDURE IN1(I); INTEGER I;  IN1♡((I-1) MOD LN)+1;
 INTEGER PROCEDURE IN2(I);     INTEGER I;
  CASE (((I-1) ◇ LN)+1 )
   BEGIN
    IN2♡-2;    IN2♡-1;      IN2♡2;      IN2♡1;
   END;
 BOOLEAN B1,B2;
 BOOLEAN ARRAY ACTIVE[1:LN4];
 LABEL STEP1,STEP2,STEP3,STEP4,STEP5,STEP6;
&
&   ♡ = REPLACEMENT      ◇ = INTEGER DIVISION
&
&   WIN= WB INVERSE
&   WIN(I,J)= W(I+1-J)  FOR  I>=J
&   P(I,J)= P(I+J(J-1)/2)  FOR  I<=J
&   IN1 AND IN2 ARE INDICATOR FUNCTIONS SUCH THAT
&    IN2(J)= -2 IF CONSTRAINT IS  I
&    IN2(J)= -1 IF CONSTRAINT IS -I
&    IN2(J)=  2 IF CONSTRAINT IS  WIN
&    IN2(J)=  1 IF CONSTRAINT IS -WIN
&    IN1(I) IS AN INDICATOR ARRAY WHICH EQUALS THE ROW NUMBER
&    OF THE CORRESPONDING IN2(I) BLOCK.
&   NQ(I) IS AN INDICATOR ARRAY WHICH POINTS TO THE ITH
&     ACTIVE CONSTRAINT
&   NORM(I)= NORM OF THE I-TH ROW OF WIN
&   ACTIVE(I)= TRUE IF THE I-TH CONSTRAINT IS ACTIVE
&   INDEX= THE COLUMN WHICH IS TO BE ADDED OR DROPPED
&
STEP1: Q♡Ø;        START♡1;
 &  COMPUTE HR=H-R
  FOR J♡1 TO (N1♡N-1) DO
```

```
   BEGIN
    T1♡-R[J];
    FOR K♡N STEP -1 UNTIL (J1♡J+1) DO
     T1♡T1+W[K]*X0[J1-K];
    HR[J]♡T1;
   END J;
  FOR J♡N TO LN DO HR[J]♡-R[J];
  &  COMPUTE Z0= WB*X0+H-R
  FOR J♡1 TO LN DO
   BEGIN
    T1♡HR[J]; J1♡J+1;
    FOR K♡(IF J<=N THEN 1 ELSE J+1-N) TO J DU
     T1♡T1+W[J1-K]*X0[K];
    Z[J]♡T1; S[J]♡-T1;
   END J;
  &  COMPUTE WIN & NORM
  NORM[1]♡T3♡WIN[1]♡1,0/W[1]; T2♡T3*T3;
  FOR J♡2 TO LN DO
   BEGIN
    T1♡0,0; J1♡J+1;
    FOR K♡(IF J<=N THEN 1 ELSE J+1-N) TO (J2♡J-1) DO
     T1♡T1+WIN[K]*W[J1-K];
    T4♡-T3*T1; WIN[J]♡T4; T2♡T2+T4*T4; NORM[J]♡SQRT(T2);
   END J;
  &  CALCULATE L
  FOR J♡1 TO LN DO
   BEGIN
    L[J]♡A[J]-R[J]; L[J+LN]♡R[J]-B[J]; T1♡0,0; J1♡J+1;
    FOR K♡1 TO J DO
     T1♡T1+WIN[J1-K]*HR[K];
    L[J+2*LN]♡C[J]+T1; L[J+3*LN]♡-(D[J]+T1);
    FOR K♡0 TO 3 DO  ACTIVE[J+K*LN]♡FALSE;
   END J;
STEP2: LSTAR♡1,0;      N1♡4*LN;
  FOR J♡1 TO N1 DO IF NOT ACTIVE[J] THEN
   &  CALCULATE LAMBDA[J] FOR ALL J NOT IN THE SET OF ACTIVE
   &              CONSTRAINTS
   BEGIN
    J1♡IN1(J);          B1♡IN2(J);
    I1♡IF ABS(B1)=1 THEN -1 ELSE 1;
    IF B1 THEN T1♡(I1*L[J]-Z[J1])/S[J1]
    ELSE
     BEGIN
      T3♡T2♡0,0; J2♡J1+1;
      FOR K♡1 TO J1 DO
       BEGIN    T4♡WIN[J2-K];
        T3♡T3+T4*Z[K];      T2♡T2+T4*S[K];
       END K;
      T1♡(I1*L[J]-T3)/T2;
     END;
    IF (T1>EPSILON) AND (T1<LSTAR) THEN
```

```
      &  RECORD THE VALUE AND INDEX OF THE MINIMUM, NON-
      &              NEGATIVE VALUE OF LAMBDA;
      BEGIN
       LSTAR♡T1; INDEX♡J;
      END;
    END J;
STEP3: IF LSTAR=1 THEN GO TO STEP6;
   &  ADD THE INDEX JUST FOUND TO THE SET OF ACTIVE CON-
   &              STRAINTS
   Q♡Q+1; START♡Q; NQ[Q]♡INDEX;
   I1♡IF ABS(IN2(INDEX))=1 THEN -LN ELSE LN;
   ACTIVE[INDEX+I1]♡ACTIVE[INDEX]♡TRUE;
   &  UPDATE Z
   FOR J♡1 TO LN DO Z[J]♡Z[J]+LSTAR*S[J];
   IF Q=1 THEN BEGIN P[1]♡1; GO TO STEP4; END;
   &  UPDATE P BY ADDING A COLUMN
   B1♡IN2(INDEX); N1♡Q-1; J3♡Q*N1◇2; I3♡IN1(INDEX); T3♡0.0;
   FOR J♡1 TO N1 DO
    BEGIN
     I1♡NQ[J];  I2♡IN1(I1); J1♡J-1; J2♡J*J1◇2;  B2♡IN2(I1);
     IF B2 THEN
      IF B1 THEN T1♡0.0
      ELSE T1♡IF I2>I3 THEN 0 ELSE WIN[I3-I2+1]/NORM[I3]
     ELSE
      IF B1 THEN T4♡IF I3>I2 THEN 0 ELSE WIN[I2-I3+1]/NORM[I2]
      ELSE
       BEGIN
        T1♡0.0; I1♡ABS(I2-I3);
        FOR K♡1 TO (IF I2<I3 THEN I2 ELSE I3) DO
         T1♡T1+WIN[K]*WIN[K+I1];
        T1♡T1/(NORM[I2]*NORM[I3]);
       END;
     IF ABS(B2*B1)=2 THEN T1♡-T1;
     FOR K♡1 TO J1 DO T1♡T1-P[K+J2]*P[K+J3];
     T2♡P[J+J3]♡T1/P[J+J2];
     T3♡T3+T2*T2;
    END J;
   P[Q+J3]♡SQRT(1.0-T3);
STEP4: FOR J♡START TO Q DO
    &  SOLVE FOR THE ELEMENTS OF BETA
    BEGIN
     I1♡NQ[J];      J1♡J-1; J2♡J*J1◇2;       T4♡L[I1];
     T1♡IF IN2(I1) THEN T4 ELSE T4/NORM[IN1(I1)];
     FOR K♡1 TO J1 DO T1♡T1-P[K+J2]*BETA[K];
     BETA[J]♡T1/P[J+J2];
    END J;
   ASTAR♡0.0;
   FOR J♡Q STEP -1 UNTIL 1 DO
    &  CALCULATE THE ELEMENTS OF ALPHA
    BEGIN
     T1♡BETA[J]; J1♡J+1;          J3♡J2♡J*J1◇2;
```

```
      FOR K♡J1 TO Q DO
       BEGIN
         T1♡T1-P[J+J2]*ALPHA[K];    J2♡J2+K
       END;
       T1♡ALPHA[J]♡T1/P[J3];
      &  FIND THE MINIMUM ELEMENT OF ALPHA AND REMEMBER ITS
      &     INDEX
      IF T1<ASTAR THEN BEGIN ASTAR♡T1; INDEX♡J; END;
     END J;
 STEP5: IF ASTAR=0 THEN
     BEGIN
      NORM2S♡NORM2Z♡0.0;
      FOR J♡1 TO LN DO
       &  CALCULATE THE ELEMENTS OF S
       BEGIN
        T1♡-Z[J]; NORM2Z♡NORM2Z+T1*T1;
        FOR K♡1 TO Q DO
         BEGIN
          I3♡NQ[K];      I2♡IN2(I3);    I1♡IN1(I3);
          T2♡IF I2 THEN IF I1=J THEN 1 ELSE 0
          ELSE IF J<=I1 THEN WIN[I1+1-J]/NORM[I1] ELSE 0;
          IF ABS(I2)=1 THEN T2♡-T2;
          T1♡T1+T2*ALPHA[K];
         END K;
        S[J]♡T1; NORM2S♡NORM2S+T1*T1;
       END J;
      IF NORM2S <XV*NORM2Z THEN GO TO STEP6
       ELSE GO TO STEP2;
     END IF;
    &  DROP THE APPROPRIATE COLUMN FROM P
    Q♡Q-1; START♡INDEX ;      I2♡NQ[INDEX];
    I1♡IF ABS(IN2(I2))=1 THEN -LN ELSE LN;
    ACTIVE[I2+I1]♡ACTIVE[I2]♡FALSE;
    IF INDEX=Q+1 THEN GO TO STEP4;
    FOR J♡INDEX TO Q DO
     BEGIN
      J1♡J+1; I1♡J2♡J1*J/2; J3♡J2-J;    NQ[J]♡NQ[J1];
      FOR K♡1 TO INDEX DO P[K+J3]♡P[K+J2];
      &  USE A GIVENS ROTATION TO ELIMINATE THE EXTRA ELEMENT
      T1♡P[J+J2]; T2♡P[J1+J2];
      P[J2]♡T3♡SQRT(T1*T1+T2*T2);
      T1♡T1/T3; T2♡T2/T3;
      FOR K♡J1 TO Q DO
       BEGIN
        I2♡I1; I1♡I1+K; T4♡P[J+I1]; T3♡P[J1+I1];
        P[J+I2]♡T1*T4+T2*T3; P[J1+I1]♡T2*T4-T1*T3;
       END K;
     END J;
    GO TO STEP4;
 STEP6: PHASE4♡(Z[1]+S[1]-HR[1])/W[1];
 END OF PHASE4;
```

REFERENCES

1. Stedman's Medical Dictionary, The Williams & Wilkins Company, Baltimore, 1966.

2. E. Corday, M.D., "Cutting the Alarming Toll of Cardiogenic Shock," Medical World News, Cardiovascular Review, 1970.

3. "Cardiogenic Shock: How To Curb It?," Medical World News, Feb 26, 1971.

4. A. C. Guyton, M.D., Textbook of Medical Physiology, W. B. Saunders Company, Philadephia, 1966.

5. W. F. Ganong, M.D., Review of Medical Physiology, Lange Medical Publications, Los Altos, California, 1967.

6. J. H. Carrington, et al., "Physical Arrangements at the Bedside in Support of Automated Systems for Patient Care," IEEE Trans. on Bio-Medical Engineering, vol. BME-18, No. 2, pp. 149-153, Mar 1971.

7. M. H. Weil, H. Shubin, D. H. Stewart, "Patient Monitoring and Intensive Care Units I," in Future Goals of Engineering and Biology and Medicine, Academic Press, pp. 232-246.

8. F. Lewis, "Patient Monitoring and Intensive Care Units II," in Future Goals of Engineering and Biology and Medicine, Academic Press, pp. 247-250.

9. J. E. Gibson, Nonlinear Automatic Control, McGraw-Hill Book Co., New York, 1963.

10. J. G. Goerner, M.D., L. A. Gerhardt, F. D. Powell, "The Application of Error Correcting Learning Machines to Linear Dynamic Systems," Proc. of the National Electronics Conference, vol. XXI, 1965.

11. P. E. Mant, "Convergent Automatic-synthesis Procedures for Sampled-data Networks with Feedback," Stanford Electronics Laboratories, SEL-64-112, TR 6773-1, Stanford, Calif., Oct 1964.

12. N. Wiener, The Extrapolation, Interpolation, and Smoothing of Stationary Time Series, MIT Press, Cambridge, Mass., 1949.

13. B. Widrow and M. E. Hoff, Jr., "Adaptive switching circuits," 1960 WESCON Conv. Rec., Inst. Radio Engnrs., Part 4, pp. 96-104. First presentation of LMS algorithm; not called this in the paper however.

14. B. Widrow, "Adaptive Filters I: Fundamentals," Rept. SEL-66-126, TR No. 6764-6, Stanford Electronics Laboratories, Stanford, Calif., Dec 1966.

15. B. Widrow, et al., "Adaptive antenna systems," Proc. IEEE, 55, Dec 1967, pp. 2143-2159.

16. L. J. Griffiths, "Signal Extraction Using Real-time Adaptation of a Linear Multichannel Filter," Rept. SEL-68-017, TR No. 6788-1, Stanford Electronics Laboratories, Stanford, Calif., Feb 1968.

17. L. J. Griffiths, "A simple adaptive algorithm for real-time processing in antenna arrays," Proc. IEEE, 57, Oct 1969, pp. 1696-1704.

18. K. D. Senne, "Adaptive Linear Discrete-time Estimation," Rept. SEL-68-090, TR No. 6778-5, Stanford Electronics Laboratories, Stanford, Calif., Jun 1968.

19. T. P. Daniell, "Adaptive Estimation with Mutually Correlated Training Samples," Rept. SEL-68-083, TR No. 6778-4, Stanford Electronics Laboratories, Stanford, Calif., Aug 1968.

20. T. P. Daniell, "Stochastic approximation for engineering application," Proc. IEEE (Letters), 57, Apr 1969, pp. 733-734.

21. J. L. Moschner, "Adaptive Filter with Clipped Input Data," Rept. SEL-70-053, TR No. 6796-1, Stanford Electronics Laboratories, Stanford, Calif., Jun 1970.

22. M. J. Installé, "A Learning Decision-Making Scheme for Use in a Time-Varying Environment," Rept. SEL-70-082, TR No. 6792-1, Stanford Electronics Laboratories, Stanford, Calif., Nov 1970.

23. J. E. Brown III, "Adaptive Estimation in Nonstationary Environments," Rept. SEL-70-056, TR No. 6795-1, Stanford Electronics Laboratories, Stanford, Calif., Aug 1970.

24. B. Goode, "Adaptive Sensor Array Processing," Ph.D. dissertation, Stanford University, Stanford, Calif., Nov 1970.

25. O. L. Frost, "Adaptive Least Squares Optimization Subject to Linear Equality Constraints," Rept. SEL-70-055, TR No. 6796-2, Stanford Electronics Laboratories, Stanford, Calif., Aug 1970.

26. J. Kaunitz, "General Purpose Hybrid Adaptive Signal Processor," Rept. SEL-71-023, TR No. 6793-2, Stanford Electronics Laboratories, Stanford, Calif., Apr 1971.

27. D. J. Austin, "Adaptive Identification and Equalization of a Digital Communication Channel," Ph.D. dissertation, Stanford University, Stanford, Calif., Jul 1971.

28. J. C. Kennedy, "Equalization of Digital Communication Channels Using BSME Criterion," Ph. D. dissertation, Stanford University, Stanford, Calif., May 1971.

29. D. S. Spain, "Identification and Modelling of Discrete, Stochastic Linear Systems," Ph.D. Dissertation, Stanford University, Stanford, Calif., Aug 1971.

30. T. Kailath, "An innovations approach to least-squares estimation, part I: linear filtering in additive white noise," IEEE Trans. Automatic Control, vol.AC-13, pp. 646-655, Dec 1968.

31. T. Kailath and P. Frost, "An innovations approach to least-squares estimation, part II: linear smoothing in additive white noise," IEEE Trans. Automatic Control, vol. AC-13, pp. 655-660, Dec 1968.

32. T. Kailath, "The innovations approach to detection and estimation theory," Proc. IEEE, 58, May 1970, pp. 680-695.

33. T. Kailath, EE 363B System Theory: Stochastic, Class Notes, Stanford University, Winter Quarter 1971.

34. A. Ralston, A First Course in Numerical Analysis, McGraw-Hill, New York, 1965.

35. F. Gantmacher, The Theory of Matrices, vol. 1, Chelsea Publishing Co., New York, 1960.

36. L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mischchenko, The Mathematical Theory of Optimal Processes, John Wiley & Sons, Inc., New York, 1969.

37. D. E. Kirk, Optimal Control Theory, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1970.

38. D. G. Luenberger, Optimization by Vector Space Methods, John Wiley & Sons, In., New York, 1969.

39. I. Flügge-Lotz, Discontinuous and Optimal Control, McGraw-Hill Book Co., New York, 1968.

40. D. Goldfarb, "Extension of Davidon's Variable Metric Method to Maximization Under Linear Inequality and Equality Constraints," SIAM J. Appl. Math., vol. 17, No. 4, Jul 1969.

41. J. B. Rosen, "The Gradient Projection Method for Nonlinear Programming, Part I -- Linear Constraints," J. Soc. Indus. App. Math., 8, 1960.

42. R. A. K. Frisch, "The Multiplex Method for Linear and Quadratic Programming," Mem. Univ. Social. Inst. of Oslo, 1957.

43. C. E. Lemke, "The Constrained Gradient Method of Linear Programming," SIAM J. Appl. Math., 9, 1961, pp. 1-17.

44. H. P. Künzi, W. Krelle, and W. Oettli, Nonlinear Programming, Blaisdell Publishing Co., Waltham, Mass., 1966.

45. E. Isaacson and H. B. Keller, Analysis of Numerical Methods, John Wiley & Sons, Inc., New York, 1966.

46. L. Kaulman, "Function Minimization and Automatic Therapeutic Control," Computer Science Report No. 228, Apr 1971.

47. P. Gill and W. Murray, "A Numerically Stable Form of the Simplex Algorithm," National Physical Laboratory Report, Maths 87, Aug 1970.

48. U. Strom, EE 280 Final Report, Sept 1971, Stanford Electronics Laboratories, Stanford University, Stanford, California.

49. N. Thompson, M.D., private correspondence.